Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Bakalářská práce

Honeypot a jeho využití v síti

Eliška Voláková

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Eliška Voláková

Systémové inženýrství a informatika
Systémové inženýrství

Název práce

Honeypot a jeho využití v síti

Název anglicky

Honeypot and its applications

Cíle práce

Bakalářská práce se zaměřuje na problematiku síťové bezpečnosti, konkrétně na monitoring podezřelých aktivit na síti, za pomoci takzvaných síťových návnad neboli honeypotů.
Hlavním cílem práce je na základě implementace honeypotu analyzovat možnosti jeho využití pro zvýšení zabezpečení počítačové sítě. Dílčí cíle jsou charakterizovat honeypot, definovat principy fungování a způsobů zaznamenávání činnosti útočníka a následný sběr dat pro výslednou analýzu.

Metodika

Metodika řešené problematiky bakalářské práce se opírá o studia a analýzy informačních zdrojů v podobě odborných a vědeckých publikací. V praktické části se realizace provede dle vlastního návrhu implementace. Následná analýza a závěr práce bude proveden na základě získaných informací z dat získaných při monitoringu.

Doporučený rozsah práce

30-40stran

Klíčová slova

Honeypot, Kippo, síťové útoky, monitoring, síťová bezpečnost

Doporučené zdroje informací

COLLINS, Michael. Network Security Through Data Analysis: Building Situational Awareness. O'Reilly Media, 2014, 348 s. ISBN 978-1449357900.

DIOGENES, Yuri a OZKAYA, Erdal. Cybersecurity: Attack and Defence Strategies. Packt Publishing Limited, 2018, 449 s. ISBN 978-1788475297.

KEONG NG, Chee, PAN, Lei a XIANG, Yang. Honeypot Frameworks and Their Applications: A New Framework [e-book]. Singapur: Springer, 2018 [cit. 2019-06-18]. e-ISBN: 978-981-10-7739-5.

MARAS, Marie-Helen. Computer Forensics: Cybercriminals, Laws, and Evidence. 2nd ed. Jones & Bartlett Learning, 2014, 408 s. ISBN 978-1449692223.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. Alexandr Vasilenko, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 3. 9. 2019

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 14. 10. 2019

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 19. 03. 2020

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Honeypot a jeho využití v síti" jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 19.3.2020 _____

Poděkování

Ráda bych touto cestou poděkovala vedoucímu práce Ing. Alexandru Vasilenkovi, Ph.D. za cenné rady, trpělivost a odborné vedení, které vedlo ke zdárnému dokončení práce.

Honeypot a jeho využití v síti

Abstrakt

Bakalářská práce se věnuje pasivní bezpečnostní technologii honeypot a jeho možnostem využití v síti. Teoretická část obsahuje komplexní přehled o dané technologii v podobě charakteristiky, a rozdělení dle míry interakce či podle specializace. Zároveň obsahuje krátký úvod do kybernetické bezpečnosti obecně.

Praktická část je věnována samotné implementaci středně interaktivního honeypotu Kippo. Obsahuje několik klíčových bodů při instalaci a konfiguraci. Především se ale zabývá samotným sběrem dat pomocí monitoringu sítě. Nejvýznamnějším prvkem praktické části je poté analýza a vyhodnocení nasbíraných dat a stanovení jistých bezpečnostních opatření. Na základě takto nasbíraných informací je proveden výstup v podobě bezpečnostních doporučení.

Klíčová slova: honeypot, SSH server, monitoring, útočník, kybernetická bezpečnost, Kippo, síťové útoky, bezpečnostní opatření

Honeypot and its applications

Abstract

The bachelor thesis deals with passive security technology honeypot and its possibilities for using in network. The theoretical part contains a comprehensive overview of the technology in the form of characteristics, and distribution by the level of interaction or by specialization. It also contains a brief introduction to cybersecurity in general.

The practical part is focused on the implementation of the medium interactive honeypot Kippo. It contains several key points during installation and configuration. Above all, however, it deals with data collection itself through network monitoring. The most important element of the practical part is then the analysis and evaluation of collected data and determination of certain security measures. Based on the information gathered in this way, an output is made in the form of safety recommendations.

Keywords: honeypot, SSH server, monitoring, attacker, cybersecurity, Kippo, network attacks, security measures

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

S příchodem největší volně propojené veřejné počítačové sítě, kterou je internet, byla většina dat masově přesunuta právě na tuto síť. Tato síť začala být využívána namísto do té doby tradičních komunikačních a jinak spojovacích prostředků. Digitalizace s sebou přinesla samozřejmě spoustu výhod, jakými jsou například dostupnost dat, rychlost komunikace či možnost jednoduchého získání obrovské škály informací z jednoho místa. Tento virtuální prostor se stal nedílnou součástí běžného života většiny populace a kompletně změnil definici pracovního procesu. Přinesl s sebou však také jisté nevýhody.

Přestože je dostupnost dat téměř odkudkoliv samozřejmě jednou z největších výhod internetu, je zároveň také jednou z největších nevýhod. A to v případě, kdy tato data začnou zajímat jisté, nejčastěji anonymní, uživatele této veřejné sítě, obecně známé pod pojmem hacker. Toto označení počítačových zločinců se však do společnosti dostalo naprosto mylně. Hacker je totiž ve své podstatě specialista, který se sice záměrně snaží najít různé mezery v zabezpečení, avšak z důvodu možného opravení těchto bezpečnostních chyb, a tím pádem zkvalitnění celkové bezpečnosti. Přesnějším termínem pro tyto uživatele je označení cracker. Tito lidé již opravdu využívají svých znalostí nejčastěji ku svému prospěchu. Tento typ kriminální činnosti se jinak nazývá také kyberkriminalita a dnes se jedná o čím dál tím častěji skloňovaný pojem.

Měli bychom si uvědomit, že každý uživatel připojený na tuto veřejnou síť, se od chvíle připojení stává potenciálním terčem útočníků. Nejčastěji jsou tyto útoky mířeny například na internetové bankovnictví, kdy se útočník snaží o získání přístupových údajů k nejrůznějším službám. Proto je nezbytně nutné, aby i tito koncoví uživatelé byli o této problematice informováni a mohli se alespoň základními kroky bránit. A to obzvláště pokud je daná osoba připojena například přes školní či zaměstnanecký účet. Pro nejméně zdatné uživatele postačí například jen samotné aktualizační verze využívaných programů a aplikací, ve kterých se jejich distributoři a vývojáři často snaží těmito bezpečnostními aktualizacemi reagovat na reálně zjištěné hrozby.

O problémech počítačové bezpečnosti slýcháme v dnešní době velice často. Existuje a dále se vyvíjí spousta technologií například jen na samotnou detekci podezřelé činnosti. Mezi tyto prvky můžeme zařadit také honeypot. Honeypot sice neslouží k samotnému

zabezpečení sítě, ale je významným nástrojem k detekci možného útoku a následnému sběru dat o útočníkových krocích. Z těchto dat je možné získat důležité informace dále vedoucí ke zkvalitnění počítačové bezpečnosti. Honeypot je přitom jednoduchá technologie, běžící na volných portech routeru. V doslovném překladu je to „hrnec medu“, který, jak název napovídá, je jakousi návnadou a zároveň pastí pro útočníky. Jeho princip je velice jednoduchý. Vytváří totiž věrnou iluzi služeb, ať už se jedná o databáze či webové stránky a podobně, čímž láká útočníky k prolomení bezpečnostních opatření, k nahlédnutí a následnému zneužití dat. Po celou dobu však honeypot sleduje jednotlivé kroky útočníka, ukládá je a následně nabízí k vyhodnocení těchto nasbíraných dat. Administrátor přitom může sledovat aktuální dění na honeypotu a snahu útočníka kdykoliv utnout.

2 Cíl práce a metodika

Dílčím cílem práce je za pomoci studia informačních zdrojů provést souhrn nejdůležitějších informací o pojmu honeypot, popis základního rozdělení dle míry interakce včetně příkladů a krátký přehled o možnostech jeho využití ve specializovaných podobách.

Praktická část je poté postavena na implementaci středně interaktivního honeypotu Kippo. Je provedena jeho implementace a spuštění. Následně je pomocí monitoringu sítě prováděn sběr dat. Hlavním cílem bakalářské práce je poté samotné vyhodnocení těchto dat a podrobení důkladné analýze, na jejímž základě je nakonec stanoveno několik doporučení týkající se bezpečnostních opatření a na závěr jsou poté uvedeny další možnosti využití takto nasbíraných dat včetně příkladu v podobě již fungujícího projektu.

3 Teoretická východiska

Cílem teoretické části bakalářské práce je krátký popis termínu počítačová bezpečnost, a to z pohledu uživatele i útočníka, ale především uvedení do technologie honeypot jakožto „návnady pro útočníky“. Doslovný překlad „hrnec medu“ naprosto vystihuje smysl tohoto bezpečnostního prostředku. Honeypot se může tvářit například jako služba, která láká útočníky k síťovému útoku. Slouží ke shromažďování informací o útočnících, vzorcích a technikách útoku. Právě díky těmto vlastnostem je honeypot velice užitečným a účinným prostředkem, jak zajistit bezpečnost sítě, a to i přesto, že do bezpečnosti nijak přímo nezasahuje. Honeypoty mají spoustu vlastností, kterými se vzájemně odlišují, a díky kterým máme možnost vybrat ten nejlepší pro naše účely.

3.1 Uživatelská bezpečnost

Ať už se hovoří o počítačové bezpečnosti v rámci organizace s rozsáhlou síťovou infrastrukturou či malou domácí sítí, téměř v každém případě nejzranitelnější část bývá nejčastěji koncový článek této sítě. Může jím být člen rodiny, ale také řadový zaměstnanec firmy. Proto je důležité dodržovat jistá pravidla. V první řadě je potřeba provádět kontrolu zásad zabezpečení. Je tedy nutná existence jisté bezpečnostní politiky, opírající se o bezpečnostní trojici, do které zařazujeme důvěrnost, integritu a dostupnost. Nejjednodušším a zároveň jedním z nejdůležitějších bodů je specifikace rolí, určení jejich práv a tím pádem také jisté odpovědnosti. Neméně důležité je potom zvyšování povědomí koncového uživatele minimálně o základech této problematiky v podobě různých školení či seminářů. Mezi časté taktiky útočníků patří rozesílání malware s fiktivními soubory v podobě faktur, které jsou dle pokynů nutné zaplatit. Uživatelé se často vyděsí, kliknou na nabízený odkaz a tím je systém kompromitován. Proto by s tímto měli být uživatelé seznámeni.

Z pohledu útočníka je první fází útoku hledání zranitelného cíle, tedy snaha o získání co nejvíce informací o celkové síti, systému, ale také třeba o zastaralých zařízeních či aktivitách zaměstnanců na sociálních sítích. Tyto informace pomáhají útočníkovi v rozhodnutí, jaké vhodné techniky při útoku využít. Nejvýhodnější je samozřejmě útok mířený přímo na někoho s jistými právy, ale může využít také slabší články tohoto systému, jimiž mohou být právě oni řadoví zaměstnanci, kteří jim jednoduše umožní vstup do sítě.

Nejčastěji využívají techniky phishing. Při využití této techniky se pomocí podvodné elektronické komunikace snaží útočník od své oběti získat například citlivé osobní údaje. V této situaci však postačí přihlašovací údaje do systému cílové organizace. S těmito informacemi se útočník jednoduše dostane dovnitř sítě. Mezi další možnosti patří nalezení chyby v zabezpečení, ať už jsou to chyby v ověřovacím kódu nebo jiné nepředvídané chyby vývojářů. Proto vývojáři neustále poskytují uživatelům různé bezpečnostní aktualizace, které reagují na pozorované, či nahlášené chyby ve spravovaných systémech. Následným krokem je skenování, tedy kritické prozkoumávání slabin systému, a hledání mezer, kterých by se dalo využít k provedení útoku. Tato fáze zabere útočníkovi značný čas, jelikož si je vědom, že tento proces může rozhodnout o úspěchu jeho činu [7].

3.2 Honeypot

Honeypoty jsou vesměs systémy, které nemají žádnou produkční hodnotu a tím pádem by s nimi nikdo neměl mít zájem navázat spojení. Jediný způsob, jak se k honeypotu dostat, není přes klasické účty, ale je nutné proniknout do něj. Proto nedojde k monitorování, dokud útočník nepronikne do nastraženého honeypotu [12]. Principem této technologie je co nejvěrnější simulace reálné sítě, snažící se nalákat útočníka tím, že se maskuje jako oblíbená zranitelná služba, ale také například aplikace, či celá síť. Tím umožňuje sledovat chování crackera a dále napomoci k vývoji nástrojů počítačové bezpečnosti. Ani honeypot není však naprosto 100% spolehlivý, a zkušený útočník jej může detekovat, jelikož informace o honeypotech a anti-honeypot nástrojích jsou již široce dostupné a umožňují útočníkům odhalit některé nastražené pasti [3]. Nejedná se tedy o konkrétní způsob ochrany sítě, ale spíše o další ochrannou vrstvu, jenž dokáže odvést pozornost od hodnotnějších prvků v síti. Častokrát, pokud útočník past prokoukne, nechce dále riskovat plýtvání časem na další možné nástraze a zaměří se na jiný cíl. Přesto je nejvhodnější honeypot naprosto oddělit od zbytku sítě, jelikož špatně nastavený honeypot může naopak posloužit útočníkovi právě jako otevřená brána do sítě [2].

Honeypot se dělí dle různých hledisek. Mezi základní rozdělení řadíme způsob použití. Jsou dva základní typy, které vycházejí z konceptu Martyho Roesche, jenž byl vývojářem Open source systému Snort na detekci neoprávněných vniknutí a prevence vniknutí. Běžně využívaný honeypot je tak zvaně produkční. Produkční honeypot slouží jako další část zabezpečení sítě umožňující mitigovat rizika. Zároveň je tento typ honeypotu

nejčastěji využívaný v komerčních organizacích, kde napomáhá právě zmírnit rizika útočníků a detekovat útoky. Tyto honeypoty mají obvykle snazší implementaci než honeypoty výzkumné, které jsou druhým základním typem této technologie, a to z důvodu jednoduchosti zapojení a nízké sofistikovanosti. Díky své relativní jednoduchosti jsou zároveň bezpečnější vzhledem k omezenému přístupu do sítě, a tedy omezené možnosti pro případné útočníky, pro které je mnohem obtížnější použít produkční honeypot k útoku a poškození jiných systémů. Nevýhodou jsou však omezené informace získané z nasbíraných dat. Výzkumné honeypoty jsou naopak navrženy na získávání co nejvíce informací o komunitě Blackhat. Mají za úkol především zkoumat jednotlivé hrozby, kterým organizace čelí, kdo jsou útočníci, jak jsou organizováni, jaké nástroje k útoku používají a kde tyto nástroje získali, což nám napomáhá lépe porozumět tomu, kdo nebo co je naší hrozbou a jak funguje. Díky těmto znalostem je možnost ochrany dat samozřejmě vyšší. Tyto více sofistikované honeypoty však přináší také své nevýhody v podobě zvýšeného rizika a větší časové náročnosti na administraci, což by se dalo naopak považovat za potenciální snížení bezpečnosti organizace. Toto rozdělení však není naprosto absolutní. Honeypot může být použit jako výzkumný i jako produkční honeypot, jelikož nezáleží na tom, jak je honeypot postaven, ale jak a pro co je využíván [2].

3.2.1 Historie

Kybernetické útoky ještě během 80. a počátku 90. let nebyly považovány za významné bezpečnostní incidenty. Do popředí se dostali až koncem 90. let, kdy se viry, červi, trojští koně a DDoS útoky vyvíjeli mnohem rychleji než virové podpisy, tedy jakási sada jedinečných bitů kódu nebo dat, které umožňují antivirovým programům jejich identifikaci, detekci a následné odstranění viru [1].

Historicky první zmínky o technologii podobné honeypotu se datují do konce 20. století, konkrétně do dob Studené války, kdy byl tento koncept vytvořen a poprvé použit jako špionážní technika. Zmiňuje se o ní Cliff Stoll v knize The Cuckoo's Egg, která pojednává o zjištění přítomnosti neoprávněného uživatele v systému. Správce systému následně 2 roky sleduje útočníka, který se snažil proniknout do více než 450 počítačových systémů. Toto pronásledování ho nakonec dovedlo až ke skupině západoněmeckých crackerů s vazbami na sovětské KGB [2][4].

Další publikací zmiňující se o této problematice je kniha Billa Chestwicka An Evening With Berferd. Snaha organizací zjistit kdo jsou jejich nepřátelé, jak mohou útočit, kdy mohou útočit, jakým způsobem zkompromitovali systém a také proč vůbec útočili, vedla v roce 1997 až k první verzi tzv. Deception Toolkit, jenž bylo jako jedno z prvních dostupné bezpečnostním technikům. Dále můžeme zmínit také projekt CyberCop z roku 1998, který byl prvním komerčním honeypotem dostupným pro veřejnost, či NetFacade4 z téhož roku [1][2].

Následně v roce 1999 vznikl pod vedením Lance Spitznera projekt složený ze skupiny lidí z celého světa zvaný The Honeynet project, jehož následné rozšíření proběhlo v roce 2002. The Honeynet project je nezisková skupina, zabývající se komunitou Blackhat, tedy crackerů, narušujících počítačovou bezpečnost ze škodolibosti, nebo k osobnímu prospěchu, či ku prospěchu třetích osob, a zároveň také výzkumem honeypotů, a sdílející své poznatky, nástroje a techniky s ostatními [1].

Honeypot se od tradičních bezpečnostních mechanismů velice liší. Je chápán jako bezpečnostní zdroj, jenž má být sondován, napaden a ohrožen. V roce 2002, kdy byl honeypot nejvíce zkoumán po rozšíření zmiňovaného The Honeynet project, byla pomocí honeypotů provedena statistika, která prokázala, že začátkem onoho roku byla domácí síť denně prohledávána 31 různými systémy [2].

3.2.2 Výhody a nevýhody honeypot

Výhody	*Nevýhody*
Shromažďování reálných dat ze skutečných útoků a dalších neautorizovaných činností, čímž poskytují bohatý zdroj užitečných informací pro následnou analýzu.	Jelikož honeypot shromažďuje informace pouze v případě útoku, tak pokud žádný pokus o útok nenastane, není z čeho provést výslednou analýzu.
Z důvodu, že pro zpracování velkých objemů dat během síťového provozu při hledání útoků nevyžadují vysoké výkonnostní prostředky, jsou celkem nízkonákladové.	Honeypot se od legitimního produkčního systému liší, proto zkušení hackeři mohou často honeypot od produkčního systému odlišit za použití tzv. fingerprinting technik, jako například skenování portů pomocí nástroje NMAP.

Dokážou zachytit škodlivou aktivitu i v případě, kdy útočník využívá šifrování.	Pokud útočník dostane podezření, že cílí na honeypot, útoku se raději vyhne.
Detekují pouze reálné výstrahy, jelikož oprávněný uživatel nemá důvod pokoušet se o přístup.	

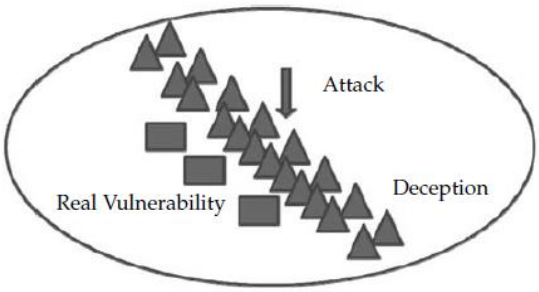
Tabulka 1 - Výhody a nevýhody honeypotu [10]

3.2.3 Statický honeypot

Honeypoty jsou v zásadě nakonfigurovány správci daného systému. Po celou dobu nasazení je přitom konfigurace statická, dokud správce neprovede konfiguraci novou. Proto je nasazení statického honeypotu snazší, jelikož není potřeba jej dále obstarávat či aktualizovat. Nevýhodou však je možnost zmeškání útoku či odhalení existence honeypotu, jelikož není schopen splynout s okolní produkční sítí. V takovém případě mohou útočníci detekovaný honeypot využít například k manipulaci a vniknutí do reálné produkční sítě. Chování honeypotu zůstává stále stejné (statické) i v případě jakékoliv změny, čímž je tento honeypot méně atraktivní a tím pádem lze přijít o zajímavé, ale především cenné informace [21]. Statický honeypot je nejjednodušší formou této technologie, přičemž může být využit jako detekční systém, ale také jako podvodný mechanismus. Statický honeypot fungující jako detekční systém simuluje falešnou zranitelnou službu, jenž má nalákat útočníka ke kompromitaci. Data, která honeypot v tomto případě nasbírá, jsou dále využívána k podrobnému prozkoumávání, analýze a vyhodnocení využívaných nástrojů a technik. Výsledky z těchto výzkumů se poté využívají ke zlepšení a doplnění funkčnosti síťových IDS pravidel, které monitorují síťový provoz, zkoumají jednotlivé pakety a snaží se v nich odhalit podezřelé aktivity a případný škodlivý kód [20].

Honeypot může mimo detekce fungovat také jako záměrně nastražená past, o které má útočník vědět. Jedná se tedy o podvodný systém, o němž útočník od začátku ví, že se jedná o honeypot. I tento způsob je však do jisté míry účinný, jelikož podezření na přítomnost honeypotu útočníka často odradí od samotného útoku, jelikož si uvědomuje, že by žádné důležité informace nezískal. Přestože se tedy v síti skutečná zranitelná místa nacházejí, přidáním dalších zranitelných míst v podobě honeypotů lze od těch reálných odvést jistou pozornost. Tento způsob oklamání chrání systém například díky zvýšené pracovní zátěži, kterou musí útočník vynaložit, aby zjistil, který z pokusů bude fungovat a který selže. Zároveň umožňuje reagovat na příchozí hrozby, a to ještě dříve, než útočník

objeví skutečnou zranitelnost reálné produkční sítě. Nejistota útočníka je vysoká a tím tyto nastražené zranitelnosti chrání ty skutečné před zneužitím. Obrázek číslo 1 poté znázorňuje síť s reálnými a nastraženými zranitelnostmi [1].



Obrázek 1 - Honeypot jako podvodný systém [1]

Mezi cíle tohoto nasazení honeypotů patří především zjištění motivu, ale také včasná detekce útoku samotného. Hlavním cílem a zároveň také smyslem tohoto nasazení je právě ono snížení pravděpodobnosti o nalezení skutečné mezery, a to díky zvětšení tohoto prostoru pomocí falešných zranitelností, jenž musí útočník prozkoumat.

Tento model nasazení získal využití jako tak zvaná sada nástrojů pro podvody (Deception Toolkit). Ta je však účinná především v případě, že se útočník stále nedostal do sítě. Proto je důležité dodržovat pravidla v podobě dostatečné velikosti tak zvaného vyhledávacího prostoru a zároveň zvýšení kvality nasazených honeypotů. Deception Toolkit totiž funguje na principu naslouchání na vstupech a poskytování zdánlivě běžných reakcí, které mají za úkol útočníka zmást nebo zdržet. Tyto odezvy jsou přitom volně programovatelné, ale vyžadují dokonalou znalost a porozumění danému protokolu [1].

3.2.4 Dynamický honeypot

Největší výzvou honeypotu je potřeba jejich konfigurace a ruční aktualizace z důvodu přizpůsobení se prostředí. Veškerá nesprávně zvolená konfigurace může vést ke zmeškání detekce, nebo obecné selhání honeypotu, a tedy ohrožení celé produkční sítě [3]. Dynamický honeypot má schopnost přizpůsobovat se a spravovat se sám za sebe. Jedná se o typ inteligentního honeypotu, ve kterém není správce nucen zavčas monitorovat a ručně konfigurovat honeypot. Kupodivu je tato verze honeypotu méně náchylná k chybám, jelikož je k nasazení potřeba méně úsilí a lidských zásahů. Nejvýznamnější výhodou je schopnost tohoto typu honeypotu se aktivně přizpůsobit aktuálnímu síťovému prostředí pomocí

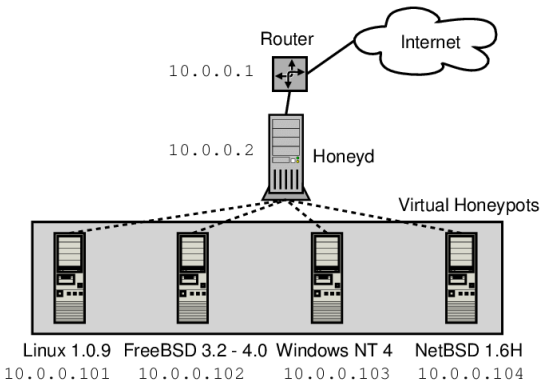
prozkoumání sítě. Následně tyto honeypoty neustále sledují případné změny, na jejichž základě aktualizují aktuální konfiguraci, například v podobě využití linuxových honeypotů v případě linuxového hostitele a podobně. Snaží se tedy monitorovat síť a průběžně shromažďovat informace, na základě kterých nasadí honeypoty v odpovídající konfigurací. Od produkčních hostitelů sbírá informace například v podobě použitého operačního systému a jeho verze, době provozu či o otevřených portech. Díky neustálému monitoringu síťového provozu dokáže identifikovat momentální atraktivní síťové služby a nasadit takové honeypoty, které tyto služby provozují, což slouží jako vhodná návnada k přilákání útočníku k interakci [21].

Zajímavé je však získávání informací o dané síti. Nejčastěji je využíváno tak zvaného aktivního a pasivního fingerprintingu. Při aktivním otisku je využíváno technik například v podobě skenování portů, fungující na principu zasílání paketů cílovému hostiteli a na základě odpovědi identifikuje roli každého jednotlivého hostitele. Dle Xuxian Jiang, jenž je docentem na katedře informatiky a jedním z hlavních členů laboratoře kybernetické obrany v Severní Karolíně v USA, by architektura takového honeypotu měla být sestavena z profilového modulu, aktivačního a deaktivačního modulu, modulu sloužícímu jakožto „továrna“ na honeypoty a modulu nasazení. Profilový modul se snaží zachytit aktivity v síti a výsledky poskytuje aktivačnímu modulu. Ten je využije k výběru vhodného operačního systému a vybrané služby. Modul s cílem vytvoření honeypotu následně vytvoří a nasadí honeypot s vybranou konfigurací. Na základě dat nasbíraných profilovým modulem se může jednat o honeypot nízko interaktivní i vysoko interaktivní. Dle implementačního modulu nasazení může být buď fyzický či virtuální. Deaktivační modul poté v případě celkového ohrožení izoluje honeypot. Výhodou tohoto nasazení je přesnější informovanost a potřeba méně času k identifikaci hostitele v síti.

Pasivní metoda fingerprintingu je pouze nasazena v síti. Zachycuje síťovou aktivitu a shromažďuje a používá získané informace k identifikaci každého hostitele. Architektura využívající pasivní otisk umožňuje nasazení v jakémkoliv prostředí pomocí samokonfigurace a to za použití již existujících honeypotu, mezi které patří Sebek či Dionaea. Nevýhodou aktivního modelu je jeho patrná přítomnost v síti, což pasivní model díky pouhému naslouchání nemá. Proto je nejvhodnější využít kombinaci těchto dvou metod a využít výhod každé z nich [3]. Další výhodou tohoto pasivního modelu je mnohem menší pravděpodobnost poškození systému či služby a samozřejmě také identifikace systému

minimálně v podobě zmapování MAC adres na adresy IP. Nevýhodou je však nedostatečná funkčnost ve směrovaných sítích, nejefektivnější je proto v sítích LAN. Po připojení tohoto honeypotu prvních 24-72 hodin pozoruje a mapuje danou síť pasivní analýzou veškerého provozu. Zajímá se o informace týkající se počtu systémů v síti, typech operačních systémů a nabízených službách, popřípadě také které systémy, a jak často s kým komunikují. Jakmile je prostředí honeypotem dostatečně zmapováno, může být nasazeno více honeypotů, jenž odráží dané prostředí. Reakce na provedené změny jej dělá více reálnějším a účinnějším. Prostředí honeypotů vypadající a chovající se jako reálné produkční prostředí umožnuje získat podezření o této pasti naprosto minimálně. Z toho vyplývá, že dynamický honeypot kromě výrazného snížení práce při konfiguraci udržuje také neustále se měnící prostředí sítě.

Rozmístění honeypotů je poté jedním z problémů nasazení tohoto modelu, jelikož fyzické nasazení nového zařízení pro každou sledovanou IP adresu připravuje tento model o jednu z důležitých výhod, kterou je minimální konfigurace a obecně zásah do daného systému. Jednodušším a efektivnějším přístupem je použití virtuálních honeypotů, jenž jsou rozmístěny a jejich veškerá údržba je prováděna jediným fyzickým zařízením. Virtuální honeypoty monitorují mimo jiné také nevyužitý prostor IP adres, což nám dává jistotu, že jakákoliv aktivita na těchto adresách je neoprávněná nebo dokonce škodlivá. Na základě mapování je poté určen počet, typ a rozmístění honeypotů nejen z pohledu používaných stanic a služeb, ale také vzhledem k poměru reálně používaných systémů. Tohoto principu dynamicky vytvářet a zavádět virtuální honeypoty zrcadlící a emulující reálnou produkční síť, využívá například Open source honeypot Honeyd [1]. Obrázek číslo 2 zobrazuje jednoduchý princip fungování konkrétně honeypotu Honeyd, který sleduje a přijímá veškerý provoz přes router a na základě toho nasazuje virtuální honeypoty [22].



Obrázek 2 - Honeypot Honeyd a vytvořené virtuální honeypoty [22]

3.2.5 Virtuální honeypot

Tyto honeypoty jsou zajímavé především pro svou rozšiřitelnost a snadnou údržbu, proto můžeme mít na jednom fyzickém stroji tisíce virtuálních honeypotů. Zároveň jejich nasazení není nikterak nákladné. K nastavení těchto virtuálních honeypotů se nejčastěji využívá VMware či Linux v uživatelském režimu, tak zvané UML (User Mode Linux). Pomocí těchto dvou nástrojů je umožněno spouštět současně více operačních systémů a jejich aplikací na jednom fyzickém zařízení, a proto je shromažďování dat mnohem snadnější. Jelikož je virtuální honeypot simulován přes zařízení reagující na síťový provoz odeslaný do virtuálního honeypotu, je potřeba mít tento stroj zasazený ve veřejné síti. Tato zařízení poté používají nejčastěji překlad síťových adres NAT [23].

Pracovní stanice využívající VMware nabízí virtualizaci především platformám Windows a Linux, ve svém virtuálním prostředí však spouští různé operační systémy. Nabízí možnost využití síťového mostu, kdy honeypotům umožňuje používat kartu daného zařízení a jeví se tak jako jakýkoliv hostitel v honeynet. V případě samostatné virtuální sítě honeypotů umožňuje síť pouze pro hostitele, kdy je provoz řízen pomocí brány firewall. Jelikož pracovní stanice VMware vytváří obrazy každého hostujícího operačního systému, je možné je jednoduše přenést na jiné zařízení, nebo pomocí zálohy obnovit honeypot do původního stavu. VMware zároveň disponuje grafickým rozhraním, díky kterému je zjednodušena instalace, konfigurace, ale i samotný provoz operačních systémů. Jelikož se jedná o komerční produkt, je neustále upgradován a má aktivní podporu. Nevýhodou je však potřeba vlastního okna pro každý virtuální stroj a omezený počet spuštěných virtuálních strojů. Jelikož se nejedná o Open source, uživatelům není umožněno provádět jakékoliv vlastní úpravy [1].

Další možností je využití tak zvaného uživatelského režimu Linuxu, což je virtuální stroj běžící pouze na Linuxu. Jedná se technicky o „linuxový port k Linuxu“. Od běžných portů na hardwarovém rozhraní definovaných procesorem se jedná spíše o port softwarového rozhraní, jenž je definován přímo Linuxem. Nejčastěji jsou využívány pro testovací účely, jelikož je lze rychle a jednoduše nastavit a následně v případě nepotřebnosti zahodit. Jedná se totiž spíše o virtuální operační systém než o virtuální stroj jako takový, které napodobují celou fyzickou platformu od procesoru po jednotlivé periferie jako například při využití VMware. Jeho výhodou je umožnění lepší komunikace s hostitelským operačním systémem

a nezávislost na jeho verzi. Umožňuje spouštět veškerý software a služby používané kterýmkoliv jiným zařízením běžícím na platformě Linux. Zároveň nabízí možnost přidávat či odebírat periférie, paměť i procesory libovolně při spuštěné instanci. Tato možnost umožňuje testovat software na hardwaru, který není fyzicky dostupný. Musí být však nakonfigurován nebo podporován [24].

Rozdílem mezi klasickým fyzickým a virtuálním honeypotem spočívá v použití softwaru, nikoliv hardwaru, k emulaci sítě. Pomocí aplikačního softwaru vytváří nové samostatné prostředí operačního systému, přičemž využívá sdílený hardware fyzického zařízení, proto může být na jednom stroji obsaženo mnoho různých virtuálních honeypotů.

Sesbíraná data by neměla být uložena lokálně v místě, kde se nachází honeypot, jelikož by je útočník mohl detekovat a přijít na to, že se jedná o honeypot. Data by mohla být následně zničena nebo naprosto ztracena. Nejúčinnější je ukládání těchto informací na vzdálený server. V případě odhalení musí útočník využít pokročilejší techniky, aby získal přístup na vzdálený server zachycující danou komunikaci. Popřípadě pozměnit systém, aby zachycoval stisky na klávesnici a snímky obrazovky a tato data vzdáleně předal dál [1].

3.3 Specializované honeypoty

Během evoluce honeypotu došlo k mnoha různým specializovaným obměnám nebo spíše zaměření této technologie, vždy jako reakce na rychle narůstající znepokojivé otázky ohledně různých hrozeb. A protože může být honeypot postaven pro velmi specifický účel, podařilo se vytvořit několik specializovaných honeypotů, jenž jsou využívány například ke sledování specifických útoků či malwaru [3]. Tímto způsobem lze honeypoty také rozdělit a následně z nich zvolit vhodný typ pro naše účely.

3.3.1 Klientský honeypot

Tento typ honeypotu se snaží nalézt škodlivé či kompromitované servery útočící na klienta. Pomocí aktivního navštěvování a komunikaci se servery, kdy se dané honeypoty tváří jako běžný počítač nebo aplikace (simulace klienta), se snaží zachytit a identifikovat škodlivé servery. Útoky na straně klienta jsou zaměřeny především na zranitelnost klientských aplikací, a to třeba v podobě emailu, webového prohlížeče či běžného kancelářského softwaru. Běžné nástroje, kterými je například firewall či antivirové

programy, fungují na principu předdefinovaného podpisu známých útoků, jenž jsou uvedeny v databázi aplikace. Významnou nevýhodou však je neschopnost detekce tak zvaných zero day útoků, tedy útoků, na které stále není v databázi definován daný podpis. Útok tedy není identifikován a tím pádem nemůže být odražen. Klientský honeypot sice neposkytuje přímé zabezpečení či ochranu, ale slouží právě pro poskytování užitečných informací, na jejichž základě se dají zpřísnit pravidla při detekci, nastavení opatření a lepší porozumění právě například těchto zero day útoků [3].

3.3.2 Serverový honeypot

První webový server byl představen koncem 70.let 20.století. Tento server byl omezen využíváním pouze značkovacího jazyka HTML. Původní webové servery sloužili pouze k procházení, tedy ke čtení, bez jakékoliv interakce mezi serverem a uživatelem. Když však byly během 90.let uživatelům zpřístupněny funkce jako nahrávání a zveřejňování příspěvků či samotné vyhledávání, bylo zavedeno CGI, tedy protokol pro komunikaci s webovým serverem, s čímž zároveň přišel první útok na tyto servery v podobě jeho zahlcení, což byly mimo jiné první známé útoky na webové aplikace [11][3].

Jak název napovídá, tento typ honeypotu pracuje na principu klasického webového serveru, přesněji napodobuje službu nebo skutečný server, ať už fyzický nebo jen virtuální. Honeypot na bázi serveru čeká, až bude odhalena jeho zranitelnost a útočník se jej pokusí ohrozit, při čemž systém začne shromažďovat informace za účelem analýzy o škodlivých činnostech. Jelikož často patří mezi nízko interaktivní honeypoty, jeho implementace je velice jednoduchá, a přesto dokáže útočníkovi nabídnout kvalitní lákavé prostředí díky své flexibilitě při přidání jiného vysoko interaktivního honeypotu. Velká flexibilita však odhaluje také jisté mezery v podobě zranitelnosti vůči útokům typu RFI, či XSS, nebo SQL injekcím.

Tento typ honeypotu pracuje na základě čtyř jednoduchých částí. Prvním z nich je, že modul vygeneruje webovou stránku, na kterou útočník cílí. Dalším bodem je generování těchto stránek jen na základě dotazů, bez použití softwaru. Odkaz je přitom dostupný klasicky z vyhledávače, a přitom je veškerá interakce mezi serverem a uživatelem logována, tedy veškeré informace o přijatých požadavcích na server ze strany útočníka jsou uchovávány. V poslední řadě filtr odděluje provoz útočníka od legitimního provozu uživatelů [3].

3.3.3 Honeytoken

Honeytoken má sice stejné vlastnosti, jako samotný honeypot, tedy snaží se nalákat útočníka tím, že se tváří jako důležitá součást systému, ale nejedná se o žádný konkrétní počítač, na kterém je honeypot běžně spuštěn [8]. Je to pouze digitální entita, kterou nevědomý útočník vnímá jako hodnotnou. Může se jednat například o jednotlivé přihlašovací údaje, či o konkrétní databázi plnou dat s informacemi o zákaznících a podobně. Než se však vygenerují data pro honeytoken, je dle autorů publikace nutné znát odpovědi na důležité otázky. Mezi ně patří, jak by měl být honeytoken konstruován. Dále pro koho je konstruován a jako informace by měly či neměly být pozměněny. Po upřesnění není problém honeytoken vygenerovat ručně, avšak celý proces bude velice zdlouhavý a časové náročný. Je však možné použít existující honeytoken generátory, které šetří práci i čas. K vytvoření se využívá dvou různých postupů. Jedním z nich je tzv. režim zmatenosti, který používá reálná data a mění pouze jejich citlivé hodnoty. Druhým módem, který je možné použít, je tzv. režim generování. Při tomto módu je honeytoken sestaven zcela sám na základě jistých pravidel, které předdefinuje sám uživatel. Data jsou v tomto případě tedy zcela smyšlená a nejčastěji se využívají v oblasti anti-phishingu [9].

3.4 Rozdělení honeypotů dle míry interakce

Mezi další významné dělení honeypotů patří tak zvaná úroveň interakce. Tento ukazatel určuje, do jaké míry umožňuje honeypot vzájemně interagovat s útočníkem. Platí to však i z druhé strany. Vysoká míra interakce, která umožňuje získat honeypotu více informací o útočníkovi, umožňuje útočníkovi získat více informací o honeypotu a tím pádem také o síti, ve které se nachází [2]. Například honeypoty s vysokou mírou interakce útočníkovi umožňují integrovat se do jeho systému, kde má k dispozici veškeré příkazy a aplikace, které by mohl jako koncový uživatel očekávat, že budou nainstalovány, a může s nimi naprosto neomezeně pracovat. Cílem je přitom zachytit co nejvíce informací o technikách útočníka. Naopak nízká míra interakce neumožňuje útočníkům takové možnosti, jelikož funkce tohoto honeypotu jsou omezené pouze na tolik, aby odhalily především zdroj neoprávněné činnosti [14].

3.4.1 Nízká interakce

Honeypoty s nízkou mírou interakce vznikly z důvodu vyvážení mezi získáváním vysoce kvalitních dat o útoku a útočníkovi a rizikem možného poškození a odpovědnosti, které by mohlo být způsobeno útočníky zneužívajícími kompromitované systémy s honeypotem. Řešením byl vývoj emulovaných honeypotů, které by se mohly pokusit napodobovat různé síťové služby, aniž by útočníkovi skutečně vystavily celý operační systém. Měly být navrženy tak, aby reagovaly alespoň základním způsobem na potenciálně škodlivé síťové vstupy, umožňující protokolovat útoky a zároveň výrazně snižovat související rizika, úsilí, nasazení a složitost správy. Tyto honeypoty začínaly jako relativně jednoduché nástroje, které při nasazení čekaly na příchozí síťová připojení a nabízely pouze omezenou nabídku služeb. Vývoj však pokračoval dál a časem vznikala řešení v podobě například Nepethes pro sběr malwaru šířícího Windows, nebo speciální nízko interakční honeypoty, jako třeba Glastopf pro webové útoky či Thug, klientský honeypot s nízkou interakcí, navržený pro aktivní procházení a hodnocení potenciálně škodlivých webových stránek [12].

Honeypoty s nízkou mírou interakce jsou založen na bázi softwaru a jsou navrženy tak, aby simulovaly jednu nebo více služeb [13]. Snadno se instalují a představují jen omezené množství služeb. Proto mohou útočníci skenovat a případně se připojit jen k několika portům. Schopnost útočníka interagovat s honeypotem je velice omezená, proto je riziko pro systém nižší, avšak získané informace o útočníkovi jsou také velmi omezené [2]. Tento typ honeypotů je výhodný a vhodný především pro snadné a levné zavádění ve velkém měřítku. Také pro minimalizaci úsilí při správě a snížení rizika potenciálního útoku, či skenování sítě a tím pádem ohrožení interních hostitelů. Dále sledování šíření malwaru v síti, studium internetových hrozeb na makro úrovni nebo poskytování výstrah v reálném čase pro vysoce automatizované útoky s malým počátečním lidským vstupem, v podobě brute force útoků nebo skenování portů [12]. Spousta firem simuluje pomocí nízko interaktivních honeypotů především protokoly jako TCP/IP, které útočníkovi umožňují domnívat se, že se připojují k reálnému systému, nikoliv do převážně statického prostředí honeypotů [15].

3.4.1.1 Dionaea

Dionaea je nízko interaktivní honeypot, který byl původně vyvinut v rámci projektu Google Summer of Code v roce 2009. Využívá skriptovací jazyk Python a je považován za nástupce Nepenthes honeypotů. Již podporuje IPv6 a TLS. Cílem tohoto honeypotu je detekce shellcodů a zachycování malware, který využívá zranitelnosti nabízených služeb, a získat kopii tohoto škodlivého softwaru. Po získání kopie malwaru umožňuje Dionaea uložit tyto soubory lokálně do textových souborů, či využít některé z externích služeb a nástrojů k analýze, jako jsou Anubis, Norman Sandbox, a jiné. Jelikož Dionaea také může obsahovat jisté zneužitelné chyby, z důvodu minimalizace dopadu běží pouze v omezeném prostřední bez administrátorských práv.

Dionaea může simulovat například protokoly FTP, který na portu 21 umožňuje vytvářet adresáře, či odesílat a stahovat soubory. Také HTTP či MSSQL nebo VoIP, kdy pouze čeká na příchozí zprávy, zaznamenává jednotlivá data a dle toho reaguje [16].

3.4.2 Střední interakce

Honeypoty se střední mírou interakce existují jako střední cesta, která se snaží kombinovat výhody nízké a vysoké interakce. Jedná se o software, který simuluje službu, ale také falešný souborový systém, se kterým může útočník skutečně pracovat. Obvykle se jedná o aplikace běžící na nějakém systému než skutečný produkční systém [13]. Jsou komplexnější, což samozřejmě zvyšuje riziko odhalení a zneužití. Mohou však shromažďovat mnohem obsáhlejší, a tedy kvalitnější informace. Na rozdíl od pouhého skenování portů je možné skutečně zachytit užitečné aktivity útočníka a zjistit, co se stane po získání přístupu k systému, jak si zvyšují svá práva, a dokonce zachytit jejich sady nástrojů. Tato vyšší úroveň interakce je sice pracnější a rizikovější, ale poskytuje velké množství informací [2].

Tyto honeypoty obsahují také jakousi vrstvu virtualizace. Jednoduše poskytují reakce, které hacker očekává, zatímco honeypot sbírá datové pole. Po přijetí oznámení o použití honeypotu bezpečnostní pracovníci extrahují kód shellu k forenzní analýze. Po podrobné analýze jsou tyto honeypoty použity k napodobení činnosti, které měl shell kód provádět. Tento typ honeypotu je již složitější, jejich nasazení je více časově náročné a pro

jejich vytvoření je třeba důkladnější znalost protokolů, aplikačních služeb a zabezpečení. Honeypoty se střední mírou interakce lze považovat za honeypoty, které byly určitým způsobem přizpůsobené tak, aby vyhovovaly potřebám organizací či jednotlivcům [1].

Často je tento typ interakce využíván pro svou rovnováhu, jelikož poskytuje mnohem menší riziko k odklonění útočníků, než při vytvoření kompletního virtualizovaného či fyzického systému, a přitom s více funkcemi než nízko interaktivní honeypot. Jsou zaměřeny především na útočníky, kteří hledají konkrétní zranitelnosti. Proto mají často sofistikovanou funkčnost k nalákání ke konkrétnímu útoku, o kterém je zájem získat více informací [15].

3.4.2.1 Kippo

Kippo je honeypot se střední mírou interakce. Jedná se totiž o software, který sice simuluje službu, se kterou může útočník navázat skutečné spojení a následně komunikovat, ale také simuluje falešný souborový systém. To je v kontrastu s nízko interaktivními honeypoty, které simulují služby pomocí softwaru, ale neposkytují žádný typ simulovaného prostředí [13].

Jedná se o skript v jazyce Python, jenž byl inspirován projektem Kojoney. Tváří se jako SSH terminál klasicky běžící na portu 22 a byl vyvinut k emulaci prostředí shellu. Zachycuje zejména brute force útoky, a to především na linuxových serverech, i když je možné jej spustit i na jiných operačních systémech, provozující například databázové servery a jiné důležité aplikace. Honeypot umožňuje měnit a libovolně upravovat prostředí, ale také souborový systém či falešné uživatelské účty nebo hesla. Jednotlivé kroky provedené v prostředí honeypotu v podobě příkazů či stažených souborů jsou ukládány a je možné je dále analyzovat. Zároveň je možné provádět otisk adresářové struktury systému, ve kterém honeypot běží, čí úprava popisu hardwaru a emulovaných síťových rozhraní [17].

3.4.3 Vysoká interakce

Honeypot s vysokou mírou interakce je ve skutečnosti nakonfigurován tak, aby co nejvěrněji odrážel produkční systém. Zároveň je navržen takovým způsobem, aby v případě zkompromitování systému nabídl útočníkovi naprostou volnost. Při implementaci tohoto typu honeypotu je třeba dbát na možnost využití systému jako pracovního prostředí při útoku proti skutečnému produkčnímu systému. Proto je nutné přijmout zvláštní opatření týkající se omezení schopnosti útočníka tento honeypot použít jako pracovní bod. Z tohoto důvodu

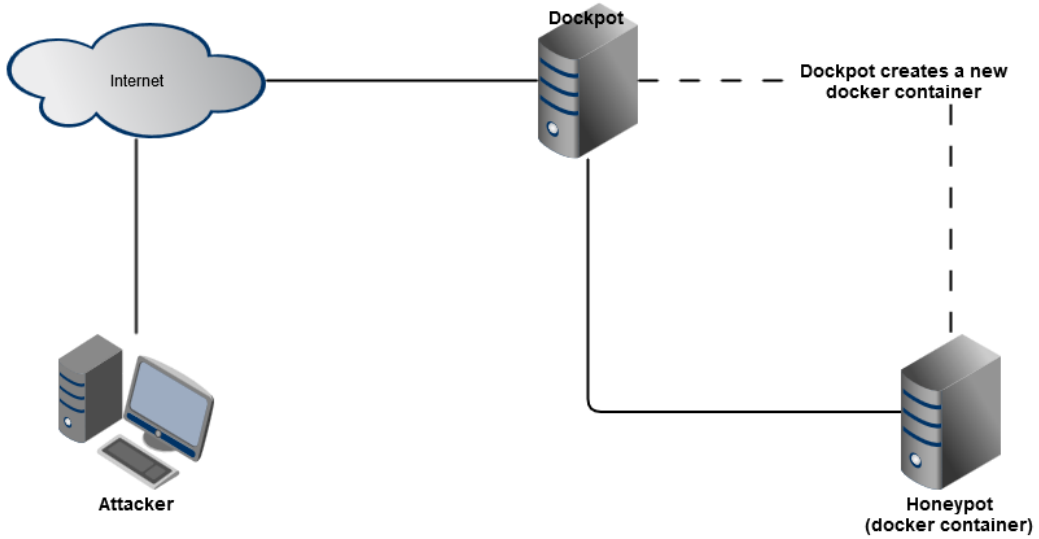
musí zároveň podléhat sadě IDS pravidel a monitoringu. Je tedy sice umožněno kompromitovat zařízení a určitým způsobem jej využívat, ale bez možnosti výhod legitimních systémů v síti. Vyžadují však spoustu práce a pečlivý přístup při jejich instalaci. Časově náročná není tedy jen instalace, ale také nutná neustálá a přísná kontrola. Tyto honeypoty jsou nejvhodnější a nejužitečnější při shromažďování zpravodajských informací, pokud se jedná o již sofistikované protivníky [13].

V případě vysoko interaktivního honeypotu se spíše než o emulaci určitého protokolu, či služby jedná o emulaci celkového reálného systému. Proto je mnohem nižší pravděpodobnost, že si útočník všimne, že byl přesměrován a je pozorován. Zároveň je díky své dynamičnosti schopný se přizpůsobit jednotlivým incidentům, a o to nižší je možnost uvědomění si, že se jedná pouze o návnadu. Díky těmto honeypotům se výzkumníci mohou dozvědět a naučit více o nástrojích použitých k eskalaci oprávnění nebo postranních úmyslů vedoucí k odhalení citlivých dat. Nejvýznamnější nevýhodou je rozhodně vložený čas a úsilí, aby i z dlouhodobého hlediska chodu honeypotu byl systém stále bezpečný a riziko nízké. Tento typ honeypotu dokáže zároveň také identifikovat opakovaně se vracející hackery, kterým přidělí jedinečné označení na základě pasivního fingerprintingu [15].

3.4.3.1 Dockpot

Dockpot je SSH honeypot s vysokou mírou interakce založený na bázi dockeru. Jedná se ve své podstatě o NAT zařízení, které je schopné působit jako SSH proxy mezi honeypotem a samotným útočníkem. Po prvním připojení na něj vytváří nové dokovací kontejnery, které obsahují potřebná data pro spuštění aplikace v podobě kódu, systémových nástrojů, knihoven, nastavení i runtime. Kontejner je zničen, jakmile je počet připojení k němu na nule. Proto není potřeba starat se o resetování zařízení s vysokou interakcí [18].

Dockpot je vesměs vylepšený honeypot HonSSH, který také stojí mezi útočníkem a honeypotem. Dockpot však vytváří nové kontejnery, kdežto vysoko-interaktivní HonSSH pouze zachycuje veškeré pokusy o připojení k textovému souboru. Jakmile se útočník pokusí přihlásit pod nějakým heslem, HonSSH nahradí tento pokus heslem správným, což útočníkovi umožňuje přihlásit se pod jakýmkoliv heslem [18] [19]. Nasazení a fungování honeypotu Dockpot je názorně vyobrazeno na obrázku číslo 3.



Obrázek 3 - Princip honeypotu Dockpot [18]

3.5 Honeynet

Honeynet jsou honeypoty s vůbec nejvyšší úrovní interakce. Princip honeynetů je naprosto jednoduchý. Jedná se o standartní síť produkčních systémů, umístěné za kontrolou přístupu například v podobě firewallu. Útočníci mohou komunikovat a využívat jakýkoliv systém uvnitř honeynetu. Běžný koncept honeypotu je založen na jediném systému, který je nakonfigurován tak, aby emuloval jiný systém nebo zranitelné místo či vytvořil naprosto uzavřené a oddělené prostředí. V případě honeynetu nejsou vytvářeny žádná prostředí ani emulované konkrétní služby, systémy v honeynetu totiž mohou být cokoliv. Dalo by se říct, že se jedná o skutečné produkční systémy.

Honeynet je běžnou fyzickou sítí více systémů, tedy typickou architekturou, kde vše poslané do této sítě je podezřelé a vše odeslané ze sítě znamená, že je síť ohrožena a útočník zahájil aktivitu. Mezi prvky definující strukturu honeynetu patří kontrola dat, zachycení dat a sběr těchto dat. Při kontrole dat by mělo být zajištěno uzavření napadeného zařízení v této síti, aby nemohlo dojít k poškození systémů, které nejsou honeypotem. Při zachycení je prováděn sběr dat z určitého zařízení. V posledním kroku poté dochází ke shromažďování údajů ze všech dat zachycených v rámci celého honeynetu [1].

Původní zastaralý koncept honeynetu z roku 1999 byl postaven na jednoduchém umístění produkčních systémů za firewall a poté pouze sledován, co se stane. Už jen toto se

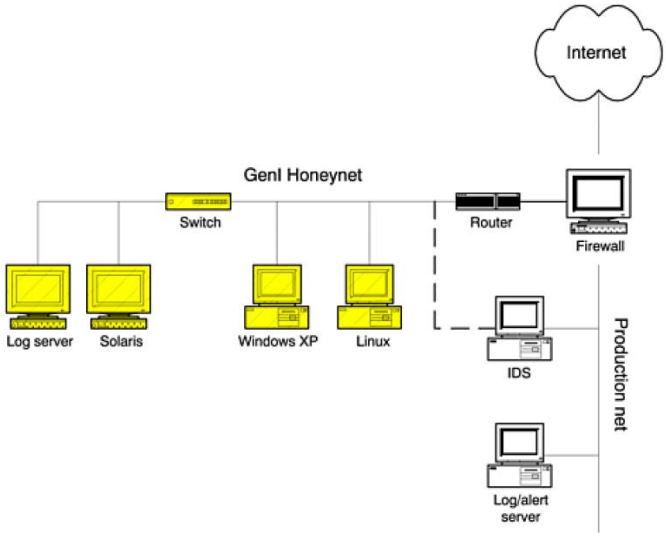
významně lišilo od tradičních honeypotů. Jedná se o velice flexibilní nástroj umožňující plnit jakoukoliv roli honeypotu. Využívají toho, že při použití skutečného systému se skutečnými aplikacemi jednodušeji útočníka oklamou. Ti mají velký problém při zjišťování, zda se opravdu jedná o reálný systém nebo pouze o návnadu. Pro svou širokou škálu využití jsou honeynety vhodné také při detekci útoků, jelikož dokážou pasivně monitorovat každý port a každý IP protokol.

Honeynet může běžet téměř na každém operačním systému, a ačkoliv se snadno dají využít jako běžné produkční honeypoty, pro svou složitost se tak jen zřídka kdy používají. Vyžadují spoustu času při samotné implementaci, ale také při běžné údržbě. Vynikají nejen v prevenci a detekci, ale také při reakci na útoky. Nejvyšší hodnotu mají při výzkumném využití, jelikož jejich hlavním smyslem a důvodem jejich vývoje je dozvědět se co nejvíce informací o hrozbách internetu. Měli by co nejvěrněji odpovídat na otázky jako kdo jsou útočníci, co je motivuje a jaké nástroje a taktiky využívají. Jakákoliv jiná implementace honeypotu není schopná získat tolik hloubkových informací jako honeynet [2].

Obecně nejdůležitější oblastí výzkumu je dozvědět se a shromažďovat co nejvíce podrobných informací o útočnících například v podobě čtení úhozů na klávesnici při kompromitaci systému, ale také komunikaci s ostatními útočníky, či nástroje používané při snímání, testování a ke zneužití zranitelných systémů. Výhodou honeynetu je možnost sledovat a učit se z těchto kroků v pro útočníky naprosto volném prostředí. Proto je lépe pochopitelné, jak fungují a proč. Jelikož honeynet nevyužívá databázi již existujících útoků, je možné díky němu objevit také nové tedy prozatím neznámé útoky. Díky tomuto honeynet vyniká také v analýze tak zvaných trendů a statistickém modelování, jelikož je na základě dlouhodobě nasbíraných dat možné sledovat a predikovat například očekávaný postup. Proto lze tento výzkumný honeypot, nebo lépe řečeno síť honeypotů, využít také k vývoji jistých obecných nástrojů, které mohou být následně aplikovány při napadení a ohrožení skutečných produkčních systémů [1].

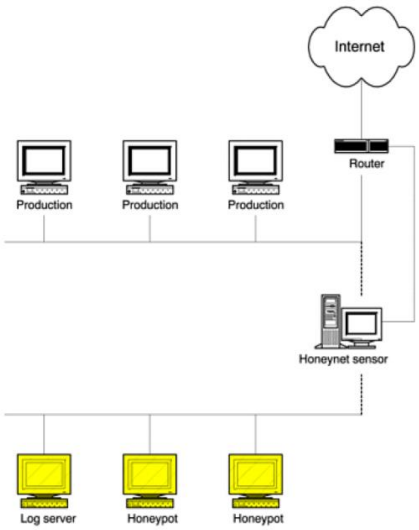
Jelikož je honeynet architekturou, nabízí se více možností jeho implementace. Mezi základní způsob zapojení patří architektury známe jako GenI a GenII. Nevýhodou GenI je relativně snadné odhalit, že se jedná pouze o návnadu. Pokročilí útočníci se tomuto tedy rovnou vyhnou. Proto většina zachycených útoků je často již známá. Jednoduše se vytvoří izolovaná síť často postavená za firewallem. Díky izolaci je sníženo riziko napadení reálné

sítě. Tato generace je využívána především na kontrolu a zachycení dat [2]. Honeynet generace I je vyobrazena na obrázku číslo 4.



Obrázek 4 - Architektura honeynet GenI [2]

Architektura GenII byla vytvořena jako řešení nevýhod GenI především v podobě zlepšení nasazení a složitější detekce. Důležitým rozdílem je například využití pouze jediného senzoru zajišťující funkci senzoru IDS v kombinaci s bránou firewall. Toto zařízení díky funkčnosti na druhé vrstvě čili stále bez použití směrování paketů, lze hůře detekovat. Z toho důvodu může být také GenII využita jako součást běžné produkční sítě [2]. Na obrázku číslo 5 je zobrazena architektura honeynetu generace 2.



Obrázek 5 - Architektura honeynet GenII [2]

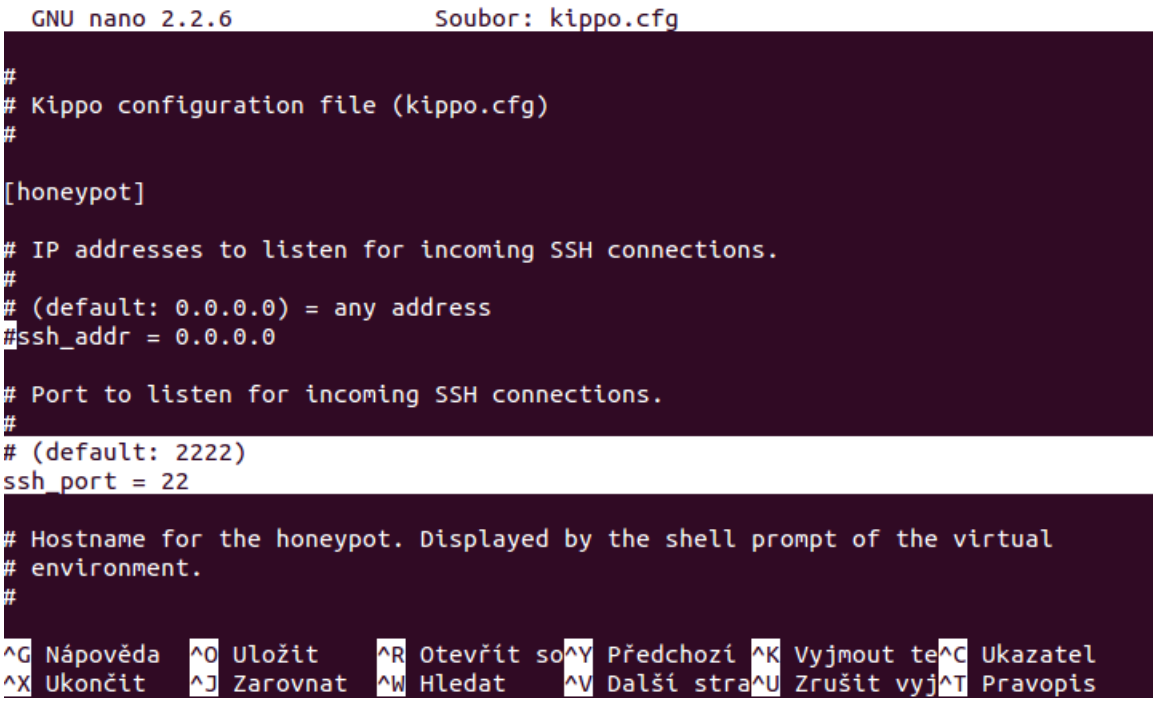
4 Vlastní práce

Praktická část práce je zaměřena na instalaci a konfiguraci honeypotu Kippo do virtuálního prostředí nástroje Oracle VM VirtualBox. Tato varianta implementace byla zvolena především z důvodu ekonomické nenáročnosti. Jedinou investicí bylo využití veřejné IP adresy, která byla poskytovatelem účtována na 99Kč, a to pouze z důvodu statického přidělení, což není nijak zvlášť nutné. Součástí vlastní práce je také sběr dat a analýza těchto dat. Na základě výsledků bude poté vyneseno několik doporučení ohledně kvalitnějšího zabezpečení sítě a zároveň provedeno porovnání s již provedenými výzkumy.

4.1 Instalace a konfigurace

V první řadě bylo nutné provést nastavení sítě v prostředí VirtualBox. Jelikož bylo potřeba, aby virtuální stroj spadal do stejné sítě jako hostitelský počítač, tedy zařízení, na němž byl VirtualBox spuštěn, využila jsem připojení pomocí NAT. V tomto případě byla hostovi, tedy virtuálnímu stroji, přidělena IP adresa z rozsahu 10.0.x.x. Aby byl však Kippo dostupný z veřejné sítě a bylo tím pádem umožněno s ním navázat spojení, bylo nutné nastavit jednoduchý port forwarding, jenž přepíná z příchozího portu 22 hostitele na port 22 hosta čili nainstalovaného honeypotu. Zároveň bylo nutné provést obdobný port forwarding na routeru, jenž umožnil přesměrování na uzel v domácí (privátní) síti LAN. Po těchto krocích byl po spuštění honeypot dostupný z veřejné sítě.

Kippo byl mimo jiné z důvodu funkčnosti pouze na starších verzích potřebných balíčků instalován na linuxové distribuci Ubuntu verze 14.04.6 (Trusty Tahr) a to na účtu nově vytvořeného uživatele server. Prvními kroky po nainstalování honeypotu bylo doinstalování potřebných balíčků k další konfiguraci a manipulaci v podobě openssh-server, openssh-client, ale také python-twisted a důležité authbind potřebné k možnosti využívání portů nižších než 1024 běžným uživatelům bez nutnosti oprávnění tak zvaného superuživatele. Z důvodu přístupu na SSH server, který představuje Kippo, bylo nutné upravit konfigurační soubor kippo.cfg a změnit ssh_port, jenž je defaultně nastaven na hodnotu 2222, na běžný port SSH serveru TCP/22. Tento krok je zobrazen na obrázku číslo 6.



Obrázek 6 - Konfigurační soubor kippo.cfg (port)

V daném konfiguračním souboru je možné pozměnit různé parametry, jenž útočníkům ztěžují identifikaci, že se nejedná o reálnou službu, ale pouze o nastraženou past. Mezi základní informace patří například změna názvu serveru, využitá verze SSH či nastavení zprávy, jenž se zobrazí při pokusu o připojení a další. V textovém souboru userdb.txt je poté potřeba nastavit přihlašovací údaje, pomocí kterých je umožněno připojit se na „server“. Dalším bodem konfigurace byla úprava souboru start.sh, kde je potřeba přepsat řádek twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid a vložit před něj authbind –deep dle obázku číslo 7.

Obrázek 7 - Změna skriptu start.sh

Tímto je instalace a konfigurace honeypotu u konce a jeho spuštění na pozadí je provedeno jednoduchým příkazem ./start.sh. Následný výpis po úspěšném spuštění skriptu je zobrazen na obrázku číslo 8.

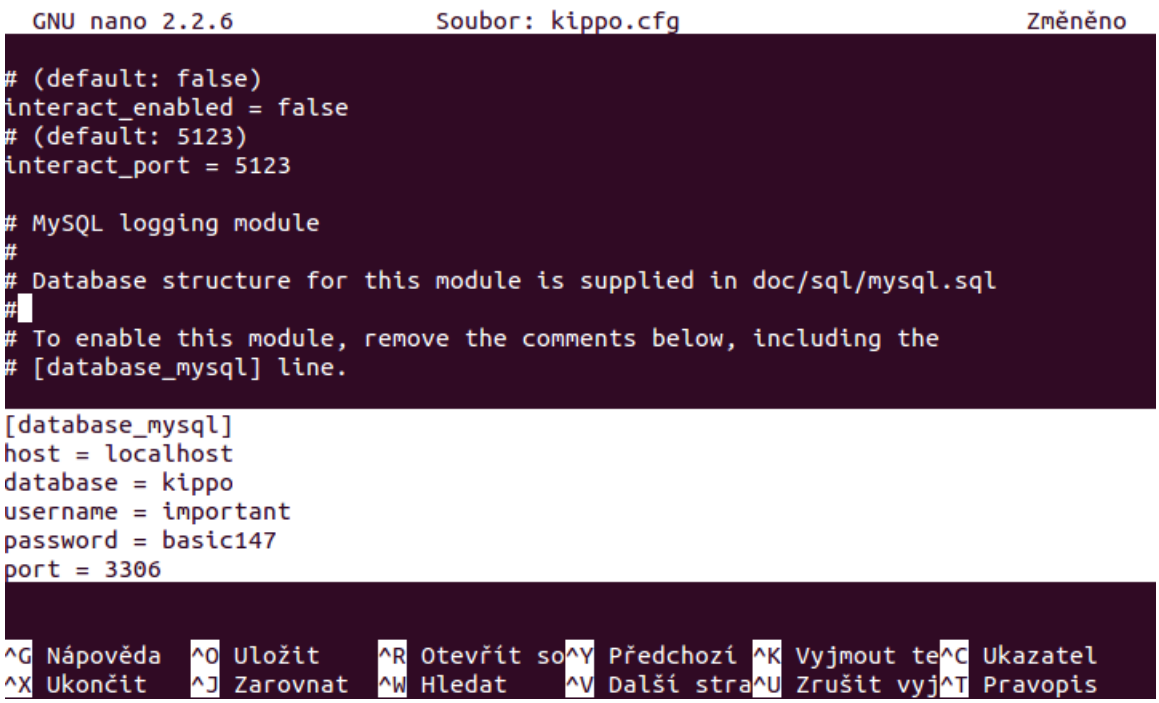
```
server@ssh-ubuntu:~/kippo$ ./start.sh
twistd (the Twisted daemon) 13.2.0
Copyright (c) 2001-2013 Twisted Matrix Laboratories.
See LICENSE for details.
Starting kippo in the background...
Removing stale pidfile /home/server/kippo/kippo.pid
server@ssh-ubuntu:~/kippo$
```

Obrázek 8 - Spuštění honeypotu na pozadí

4.1.1 Kippo-graph

Pro vizualizaci a kvalitnější a zároveň snadnější hodnocení výsledků byl využit skript kippo-graph, jenž umožňuje zobrazení statistik honeypotu Kippo. Současná a také využitá verze zobrazuje na 24 grafů, do kterých spadá například 10 nejpoužívanějších hesel, 10 nejpoužívanějších přihlašovacích uživatelských jmen a různé jejich kombinace, ale také nejvyužívanější SSH klienti či jednoduché statistiky připojení v určité dny a podobně [25].

Postup a popis instalace je snadno dohledatelný. Pro získání výstupů ze skriptu je však ještě nutná instalace mysql. V první řadě byla vytvořena mysql databáze s názvem kippo a zároveň také nový uživatel important, jemuž byla na danou databázi přidělena všechna práva. Následně byl upraven soubor config.php na základě nově vytvořené databáze a uživatele. Stejným způsobem musel být upraven konfigurační soubor kippo.cfg, konkrétně jeho část týkající se databáze mysql, jenž je zobrazen na obrázku číslo 9. V této práci bylo kvůli možnosti lepšího sledování jednotlivých zápisů využito ještě bezplatného nástroje phpMyAdmin, a to zejména pro své intuitivní webové rozhraní.

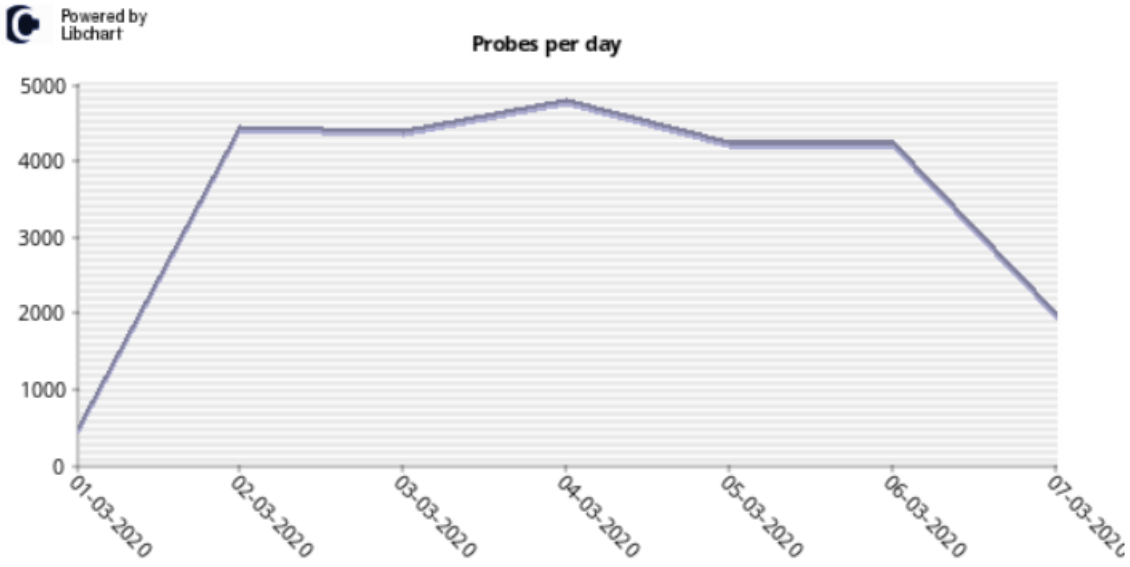


Obrázek 9 - Nastavení databáze v kippo.cfg

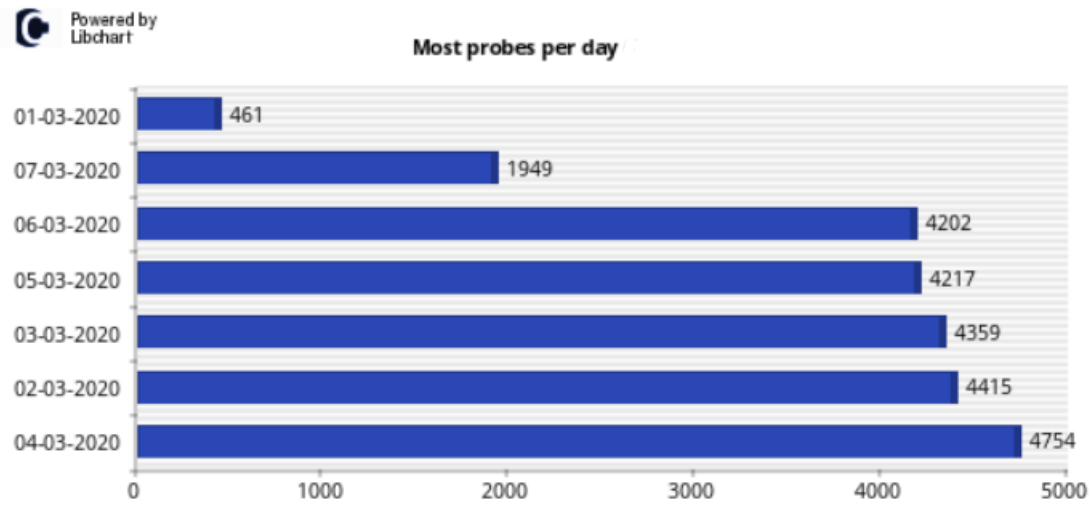
Při otevření phpMyAdmin byly do vytvořené databáze kippo importovány tabulky ze souboru mysql.sql, jenž je součástí nainstalovaného balíčku kippo. Po spuštění honeypotu jsou poté data sbírána do těchto tabulek a následně zobrazována v podobě grafů ze skriptu kippo-graph.

4.2 Analýza nasbíraných dat

Kippo-graph nabídl širokou škálu informací týkajících se pokusů o připojení na honeypot. Oficiální spuštění připravené pasti proběhlo 1.3.2020. První pokusy o přístup se objevily po 3 hodinách chodu. Pomalý nástup byl přisuzován mimo jiné tomu, že spuštění proběhlo v neděli. Tuto domněnku podpořil fakt, že následující den v ranních hodinách byl zaznamenán rapidní nárůst pokusů na téměř 1900. Že je den v týdnu klíčový, jestliže se zaměřujeme na počet pokusů, napovídá také průběh spojnicového grafu na obrázku číslo 10, jenž vyobrazuje postupné výkyvy v počtech pokusů o přístup v průběhu jednoho týdne (neděle 1.3.2020 – sobota 7.3.2020) a také graf z obrázku číslo 11, ve kterém jsou poté seřazeny jednotlivé dny v týdnu dle počtu pokusů o přístup seshora od nejméně vytíženého dne po nejvíce vytížený. Z tohoto grafu je zřejmé, že neděle 1.3.2020 byla naprosto nejslabším dnem s počtem 461 pokusy, oproti středě 4.3.2020, kdy byl počet pokusů naopak maximální při počtu 4754. Během pracovního týdne od pondělí do pátku se počet pokusů držel v rozmezí cca 4200-4800 denně.

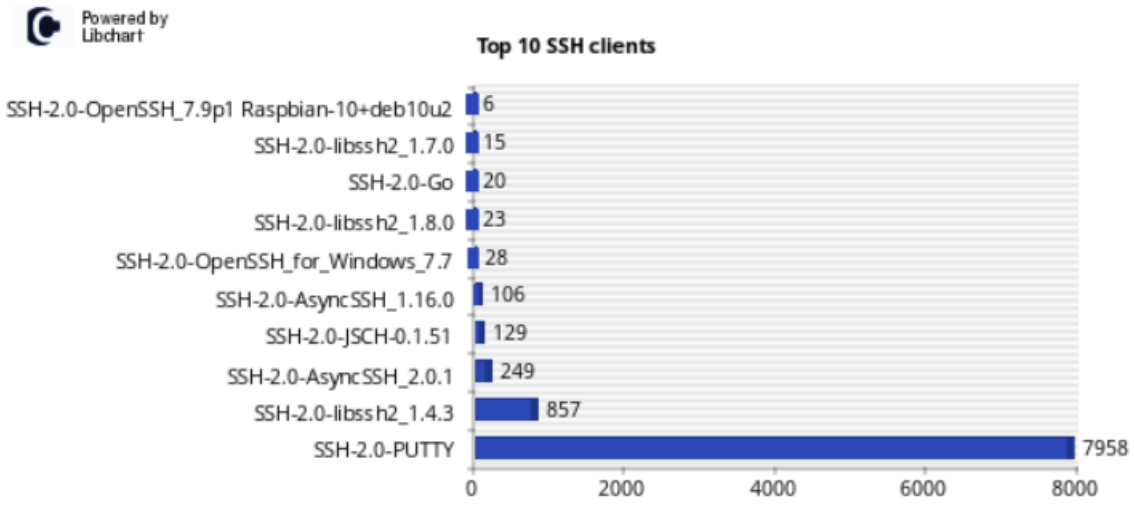


Obrázek 10 - Křivka počtu pokusů o přístup



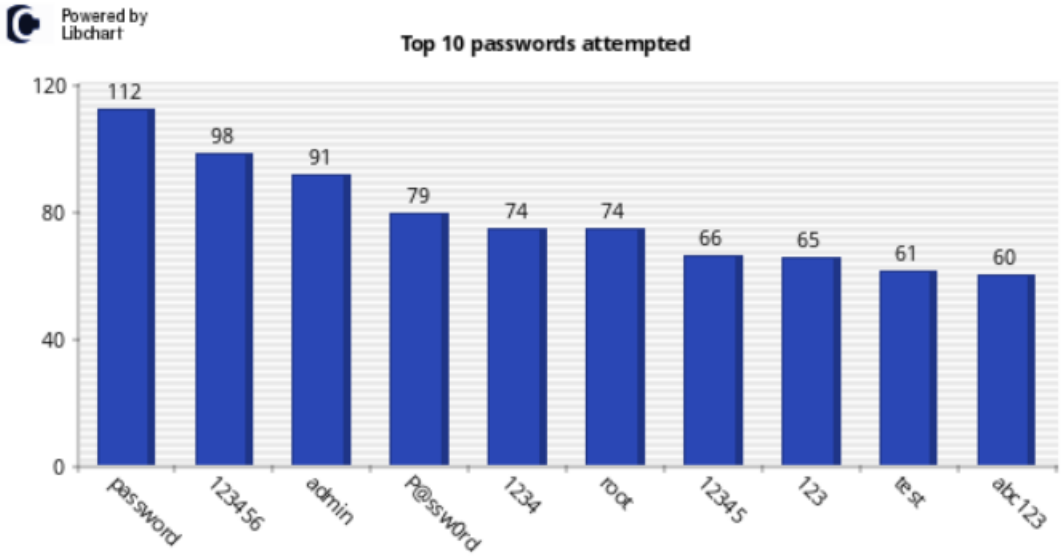
Obrázek 11 - Počet pokusů za den

Další graf z obrázku číslo 12 zobrazuje top 10 SSH klientů, kteří byli nejhojněji využívány při pokusech o přístup. Tomuto grafu naprosto dominuje SSH klient PuTTY, jenž sklidil v roce 2020 3.místo v seznamu nejčastěji využívaných SSH klientů na platformě Windows. A to dle výsledků hodnocení uživatelů pod názvem „What are the best SSH clients for Windows?“ na webu www.slant.co, jenž provádí hodnocení různých především IT produktů [26]. Klient PuTTY je mimo jiné doporučován také pro Linuxové platformy na stránkách www.pickaweb.co [27]. Jedním z důvodů je jistě také to, že kromě multiplatformnosti, kterou nově klient PuTTY disponuje, se jedná o bezplatný software. Druhé místo poté dle grafu obsadil klient libSSH2, následně AsyncSSH a v blízkém závěsu poté Java Secure Channel.



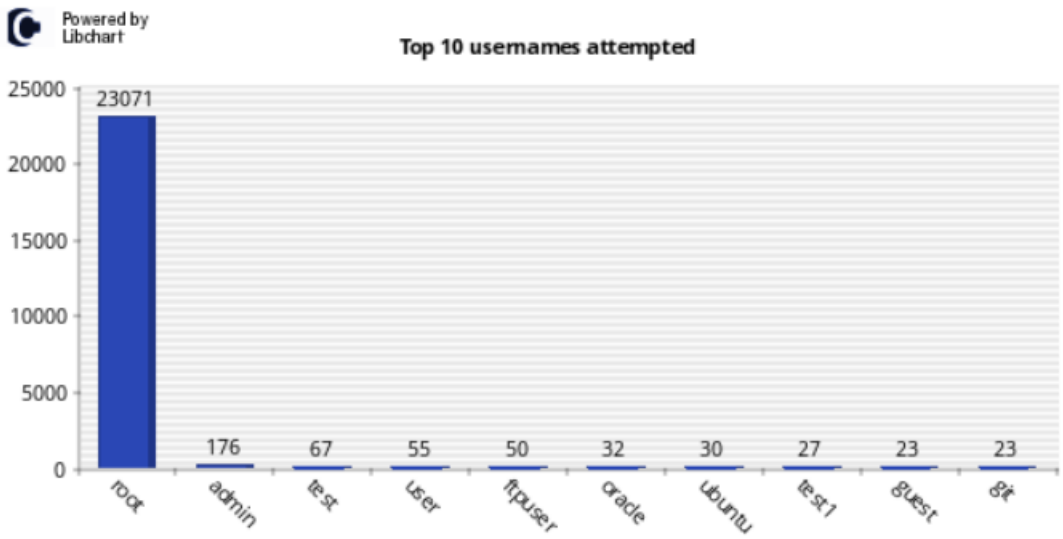
Obrázek 12 - 10 nejpoužívanějších SSH klientů

Následující graf na obrázku číslo 13 zobrazuje nejčastěji využívaná hesla při pokusech o přihlášení. Kromě očekávaných hesel v podobě password, admin, či různé kombinace čísel, které jsou často defaultním nastavením, sklidilo velký úspěch heslo P@ssw0rd, které už kombinuje speciální znaky s velkými písmeny a čísly. Mezi ne příliš běžná, avšak zajímavá hesla patří například 123qwe!@#; p0o9i8u7y6t5r4e3w2q1; !@#qazwsxedc.



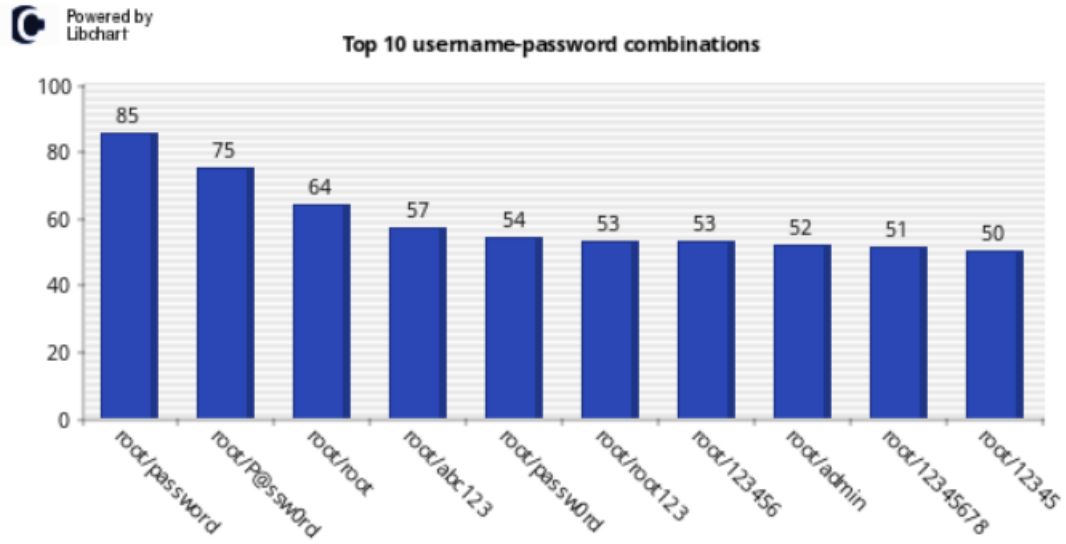
Obrázek 13 - Nejčastěji používaná hesla

Výsledky grafu z obrázku číslo 14 uvádí, že nejčastěji bylo při pokusu o spojení využito přihlašovacího jména root, a to ve 23071 případech z celkových 24326. Tento výsledek se dal v celku předpokládat vzhledem k tomu, že se jedná o defaultní název účtu administrátora v operačních systémech na bázi unixu. Na druhém místě se poté s o mnoho méně hlasy umístil uživatel admin. Společnost Rapid7, provádějící výzkumy v oblasti IT bezpečnosti, v roce 2014 spustila projekt s názvem Heisenberg, který spočívá v nasazení nízko interaktivních honeypotů. A to za účelem zjištění informací ohledně aktivit útočníků, ale i výzkumníků a organizací, a také ve snaze porozumět postupům, technikám a taktikám, které lidští i robotičtí útočníci využívají. Dle dat z roku 2016 v tomto projektu bylo pomocí RDP protokolu společnosti Microsoft zjištěno, že uživatelská jména jako admin (popřípadě Admin), či administrator (nebo Administrator) jsou nejčastěji zkoušenými uživatelskými jmény, a to v téměř 62% pokusů [28].

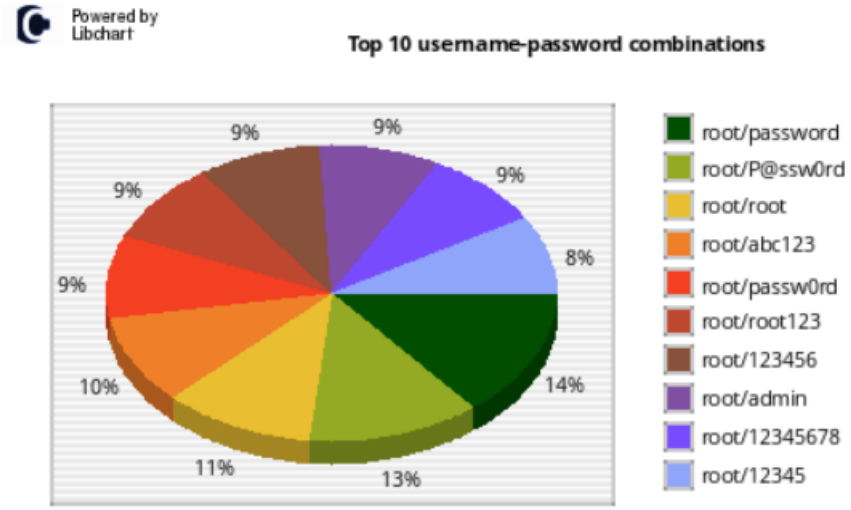


Obrázek 14 - Nejčastěji používaná uživatelská jména

Následující obrázky se sloupcovým grafem vyobrazeným na obrázku číslo 15 a pro lepší představu také výsečovým grafem z obrázku číslo 16, zobrazují nejčastější kombinace jmen a hesel použitých při pokusech o přístup. Ve všech případech bylo jako uživatelské jméno použito root, což podpořil také již zmíněný graf z obrázku číslo 14 s nejčastěji používanými uživatelskými jmény. Některá použitá hesla jsou poté také samostatně zobrazena v grafu v obrázku číslo 13.

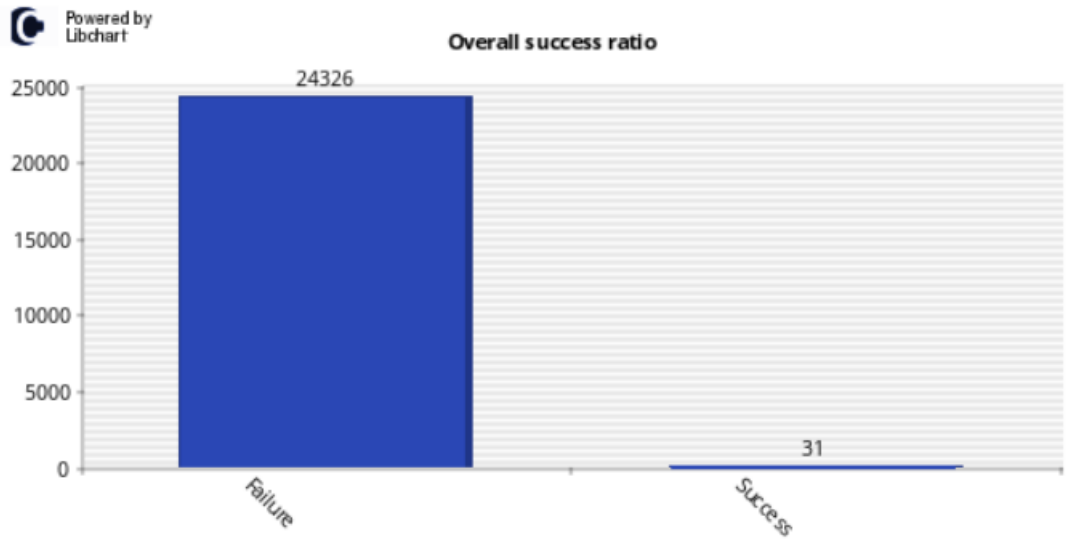


Obrázek 15 - Nejčastější kombinace uživatelského jména / hesla



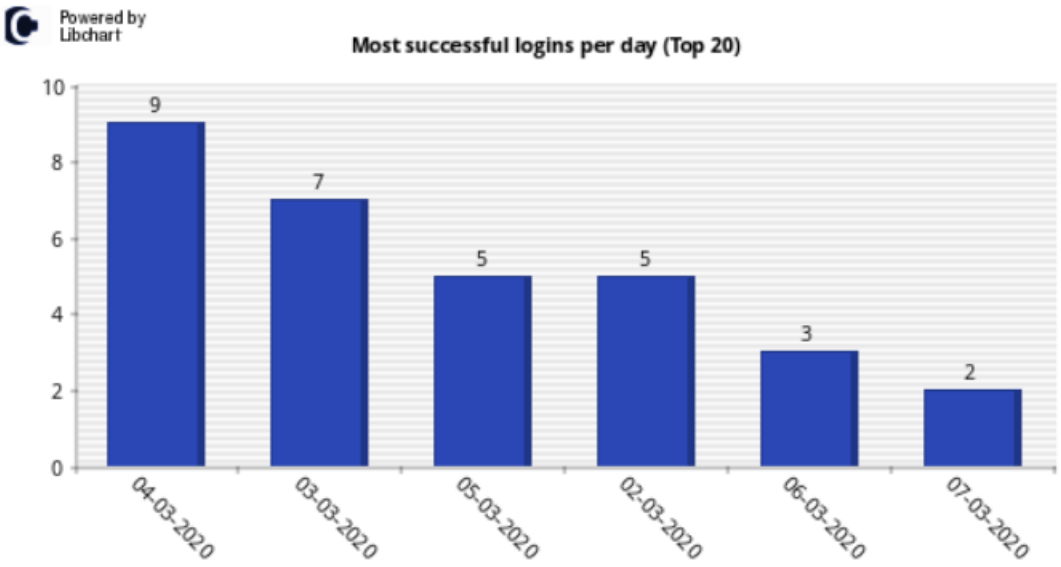
Obrázek 16 - Kombinace uživatelského jména / hesla (sloupcový graf)

Úspěšné přihlášení bylo zaznamenáno pouze v 31 případech, což je oproti celkovému počtu 24326 pokusů naprosto zanedbatelné. Poměr těchto hodnot je zaznamenán v grafu na obrázku číslo 17.



Obrázek 17 - Poměr úspěšných / neúspěšných pokusů o přihlášení

Pro zajímavost graf z obrázku číslo 18 zobrazuje počet úspěšných pokusů v daný den. Z grafu je zřejmé, že při prvním dnu, který byl obecně velmi slabý, neproběhlo ani jedno úspěšné přihlášení. Sobotu z celkového počtu 1948 pokusů byly úspěšné hned 2 přístupy. Nejvíce zdárných pokusů bylo provedeno ve středu 4.3.2020, což byl také den sčítající obecně nejvíce pokusů o přístup.



Obrázek 18 - Úspěšné přihlášení v dané dny

Tyto pokusy však končily převážně okamžitým odhlášením a ukončením celkové interakce pomocí příkazu exit. Jelikož je kippo honeypot se střední mírou interakce, obsahuje různé chyby a mezery, dle kterých se dá identifikovat, že se jedná o honeypot. Tomu nasvědčuje například využití příkazu file /sbin/init, kdy běžný systém vypíše chybovou hlášku v podobě:

```
/sbin/init: ELF 64-bit LSB shared object, x86-64, version 1
(SYSV), dynamically linked (uses shared libs), for GNU/Linux
2.6.24,
BuildID[sha1]=7a4c688d009fc1f06ffc692f5f42ab09e68582b2,
stripped
```

Kdežto Kippo odpoví jen:

```
bash: file: command not found
```

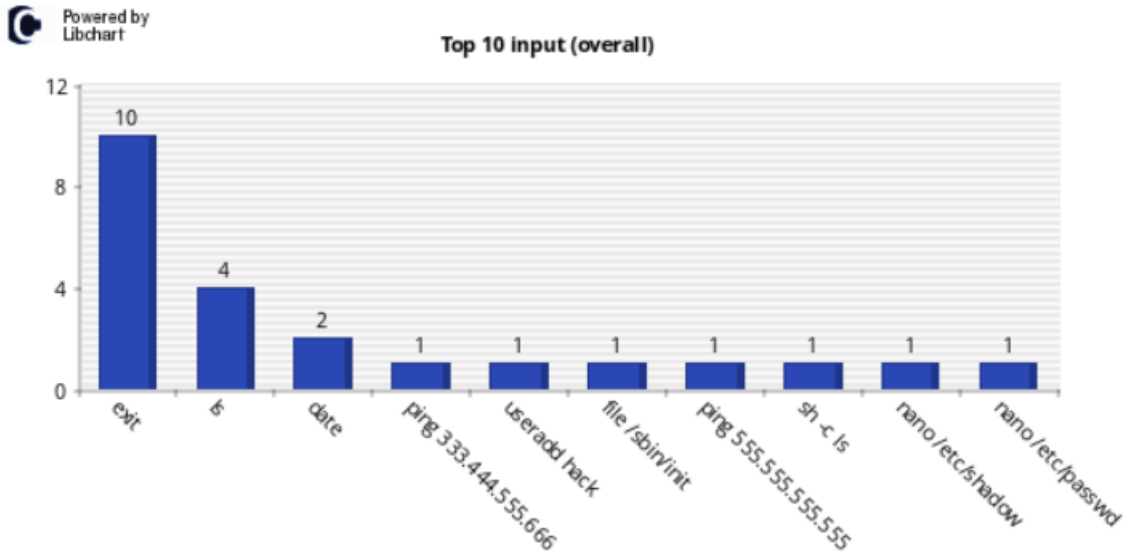
Dalším příkazem pravděpodobně zkoušejícím pouze odpověď programu, také z důvodu identifikace daného systému, byl obyčejný ping na neexistující adresu 333.444.555.666. Odpověď regulérního produkčního systému je:

```
unknown host 333.444.555.666
```

Kippo však zareaguje:

```
PING 333.444.555.666 (333.444.555.666) 56(84) bytes of data.
64 bytes from 333.444.555.666 (333.444.555.666): icmp_seq=1
ttl=50 time=46.4 ms
64 bytes from 333.444.555.666 (333.444.555.666): icmp_seq=2
ttl=50 time=48.3 ms
64 bytes from 333.444.555.666 (333.444.555.666): icmp_seq=3
ttl=50 time=48.4 ms
64 bytes from 333.444.555.666 (333.444.555.666): icmp_seq=4
ttl=50 time=45.0 ms
64 bytes from 333.444.555.666 (333.444.555.666): icmp_seq=5
ttl=50 time=47.9 ms
--- 333.444.555.666 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 907ms
rtt min/avg/max/mdev = 48.264/50.352/52.441/2.100 ms
```

Další nápovědou k identifikaci honeypotu může být také doba reakce, která se pohybuje okolo 40-50ms, kdežto reálný systém má běžnou dobu reakce okolo 5-7ms. Dané příkazy použité při úspěšném přihlášení jsou znázorněny na obrázku číslo 19.



Obrázek 19 - Nejpoužívanější příkazy při úspěšném přihlášení

Pomocí příkazů nano /etc/shadow či nano /etc/passwd se útočník snažil dostat do souborů obsahujících informace o uživatelích a provést tam nějaké změny, jedinou odpověď, kterou však od honeypotu mohl získat, bylo:

```
bash: nano: command not found
```

Tato odpověď ho určitě také varovala před tím, že se jedná pouze o past, nikoliv SSH server s reálnými daty.

4.3 Bezpečnostní opatření

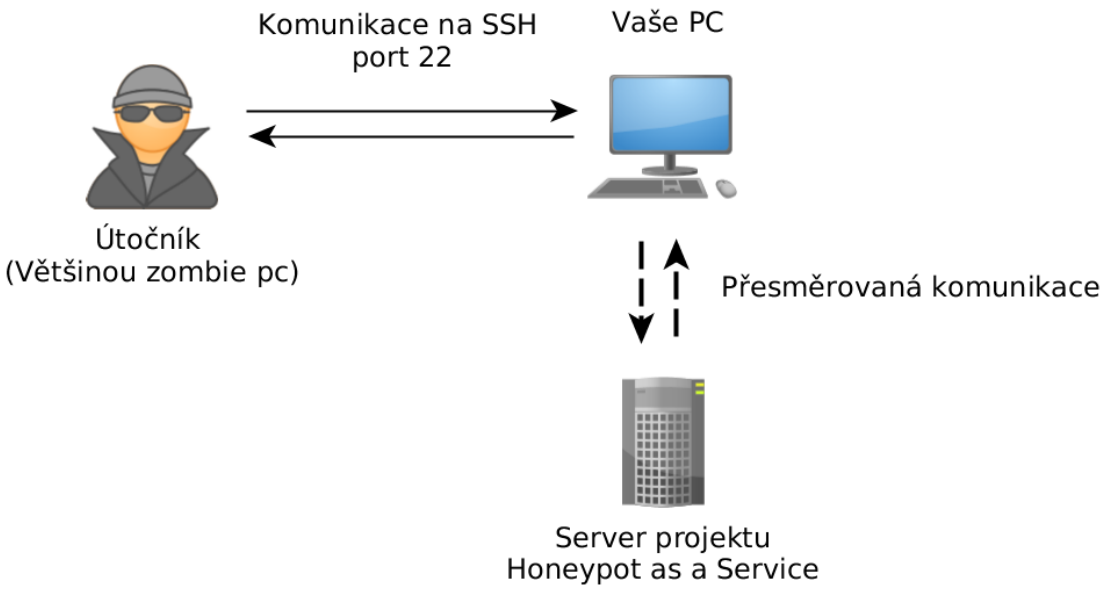
Je důležité dbát na to, aby přihlašovací jméno, ale převážně heslo byly co nejvíce bezpečné a bylo využito různých kombinací speciálních znaků, čísel a velkých a malých písmen. Výsledky z monitoringu napovídají, že daný útočník předpokládá a věří, že existuje určitá šance, že dané heslo, ale i dané kombinace jména a hesla opravdu může zafungovat. Tomu napovídá i fakt, že databáze těchto programů, které provádějí tyto skeny, jsou mimo jiné tvořeny také z uživatelských jmen a hesel, jenž se jim již podařilo prolomit. Doporučeních ohledně zkvalitnění zabezpečení je na základě monitoringu hned několik.

- Samozřejmě základem je silné uživatelské heslo k účtům. V základu je známo, a dle výsledků bylo také prokázáno, že by mělo obsahovat co nejvíce kombinací velkých a malých písmen, čísel a speciálních znaků. Velmi dobře známé, a tedy již předpokládané, jsou záměny písmen za znaky („0“ místo „o“, „$“ místo „s“ a podobně) čemuž napovídá například již zmíněné heslo P@ssw0rd, ale také využití osobních údajů nebo rozšíření přihlašovacího jména v heslu (uživatel PetrNov – heslo „novakpetr123“). Proto je lepší se takovým kombinacím vyhnout.
- Vzhledem k výsledkům monitoringu by bylo vhodné zakázat přístup minimálně pomocí uživatele root, který byl využit při pokusech o přihlášení 23071krát.
- Nejvhodnějším řešením by bylo využít přihlašování ke službám pomocí veřejného klíče, čímž odpadá starost týkající se dostatečně bezpečného hesla.
- Částečně pomoci může i změna původního (továrního) portu na port jiný. Celkový počet pokusů se za týdenní sledování vyšplhal na 24326. Jedním z důvodů tohoto enormního čísla je také to, že SSH server běžně naslouchá na daném portu 22 a tento port byl poté využit i v této práci. Změna portu sice není dostatečným zabezpečením, které by zabránilo útoku, ale minimálně sníží počet pokusů.
- Z důvodu sledování aktuálních hrozeb a mezer v zabezpečení dané sítě bych osobně doporučila monitorovat síť preventivně, a to jako pasivní součást zabezpečení. Popřípadě je možné se připojit k některému z dostupných projektů, jenž se touto tematikou zabývají. Příkladem může být například Honeypot as a Service.

4.3.1 Honeypot as a Service

V návaznosti na poslední bod bezpečnostních opatření bych se ráda zmínila o projektu nesoucí název Honeypot as a Service (HaaS), jehož beta verze vznikla říjnu roku 2017 a oficiálně byl poté spuštěn během února 2018. Projekt funguje pod záštitou sdružení CZ.NIC, které je zároveň správce domény .cz.

Princip tohoto projektu je zcela jednoduchý a zapojení se do něj je bezplatné a zcela dobrovolné. Zájemce pouze provede registraci a do svého zařízení nainstaluje proxy dostupnou na stránkách projektu. Tato proxy začne po spuštění veškerou příchozí komunikaci z portu 22 přeposílat na server HaaS, kde je implementován honeypot cowrie, jenž simuluje zařízení a jednotlivě zaznamenává provedené příkazy. Zapojené zařízení je přitom zcela v bezpečí, jelikož je veškerý provoz přesměrován na server projektu. Schéma komunikace je zobrazeno na obrázku číslo 20.



Obrázek 20 - Schéma komunikace [29]

Jediný problém nastává v případě, pokud je port 22 k přístupu k routeru již využíván. V tuto chvíli je pouze potřeba provést jednoduché přesměrování. Zapojením se do projektu uživatel přispívá ke zlepšení kybernetické bezpečnosti, ale také připravenosti na kybernetické útoky, a zároveň získává zajímavé informace o útocích provedených skrze jeho zařízení. Má přístup například k útočníkovým IP adresám, jaké ověření k připojení použil, informace o jeho chování a o skriptech, které útočník v honeypotu spustil [29].

5 Výsledky a diskuse

Přestože byl přístup na spuštěný honeypot během 7 dní funkčnosti z celkem 24326 pokusů pouze 31krát úspěšně prolomen, nebyl nastražený „server“ až na několik jednoduchých příkazů, které spíše ověřovaly pravost zařízení, nijak kompromitován. Zajímavý byl například případ uživatele využívající SSH honeypot, který dle analýzy logu zjistil, že se útočník pokusil jeho zařízení využít k těžbě kryptoměny. Uživatel využívá router Turris. Ty jsou vyvíjeny sdružením CZ.NIC, které vede zmíněný projekt HaaS [30].

Výsledky práce přesto mohou být nadále využity. Dá se předpokládat, že nejčastěji se pokoušely na honeypot útočit převážně boti, jenž běžně hledají na veřejné síti otevřené porty a snaží se pomocí kombinací jména a hesla prolomit toto zabezpečení a dostat se do systému. Na honeypot se během týdne pokoušeli uživatelé i boti přihlásit pomocí 6171 různých hesel. Jak již bylo zmíněno jako součást bezpečnostního doporučení, nejvhodnějším způsobem ochrany vůči tomuto prolomení je využívat k přihlašování veřejný klíč. V takovém případě je v tomto ohledu server sice zabezpečený, ale je dále opakovaně zahlcován. Boti bohužel často nereagují na hlášku „Permission denied (publickey).“ a zbytečně svými pokusy plní logy.

Takto získaná data využívá například Fail2ban, který prohledává autentizační logy a na základě toho blokuje přístupy z různých IP adres. V konfiguračním souboru je možné nastavit mimo jiné to, jaké IP adresy mají být ignorovány, na jak dlouho je udělen zákaz (v sekundách), dále povolený počet pokusů, než dojde k zákazu a také určitý časový limit, po který je z určité IP adresy sledována snaha o navázání spojení. Tohoto démona lze nastavit a může být využit stejně tak na SSH jako na PHP, Apache a jiné další služby. Veškerý provoz je poté zaznamenáván v logu démona. Na podobném principu funguje také skript DenyHost, který je však nakonfigurovaný pouze pro SSH.

Nejdelší doba připojení k serveru byla 382 sekund, což je více než 5 minut. Během této doby bylo spojení sice aktivní, ale nic se nedělo a za celou dobu přihlášení nebyl proveden jediný příkaz. Jako další krok pro zkvalitnění zabezpečení by měl být nakonfigurován časový interval, který po určité době nečinnosti uživatele ze serveru automaticky odhlásí. Tento interval může být nastaven libovolně dlouho dobu, která po dovršení ukončí spojení.

6 Závěr

Nejdůležitější částí bakalářské práce byla především část vlastní práce, týkající se implementace a vhodné a správné konfigurace honeypotu. Hlavním cílem poté byla analýza dat, která mohla být provedena díky monitoringu a sběru dat, a na jejímž základě měly být a byly stanoveny jistá opatření vhodná ke kvalitnějšímu zabezpečení sítě. Tato bezpečnostní doporučení jsou součástí výsledků praktické části. Finanční náročnost implementace je dá se říct minimální, až nulová, a přesto výsledky prokázaly, že tato technologie může být vhodným pomocníkem při snaze co nejlépe zabezpečit síť. S takto nasbíranými daty se může nadále pracovat například pro rozvoj a aktualizace IDS či IPS systémů.

Dílčím cílem byla teoretická část práce, kde jsem na základě informací získaných z odborných informačních zdrojů shrnula základní poznatky o pojmu honeypot. Teoretická část obsahuje krátký pohled do historie vzniku této technologie jakožto špionážní techniky, následně rozdělení dle různých aspektů, ať už se jedná o míru interakce se zmíněnými příklady v jednotlivých typech interakce, nebo dle specializovaného využití ve formě klientských či serverových honeypotů, až po rozšíření těchto návnad do širších rozměrů v podobě honeynet.

Téma práce bylo zvoleno mimo jiné pro získání nových informací v oblasti prevence a základního zabezpečení sítě. Získaná data však předčila mé očekávání. Počet pokusů cílených na bezcenný SSH server, jenž vzrostl za týden na téměř 25000, je opravdu enormní a poukazuje na důležitost této problematiky, a pozornost, jenž by měla být této oblasti věnována. Zároveň si myslím, že práce může být použita jako podklad pro další výzkumy.

7 Seznam použitých zdrojů

[1] JOSHI R.C. a SARDANA Anjali, ed. *Honeypots: A New Paradigm to Information Security*. United States: Taylor & Francis, 2011, 340 s. ISBN 978-1578087082.
[2] SPITZNER Lance. *Honeypots tracking hackers*. Boston: Addison-Wesley, 2003, 480 s. ISBN 978-0321108951.
[3] KEONG NG Chee, PAN Lei a XIANG Yang. *Honeypot Frameworks and Their Applications: A New Framework* [e-book]. Singapur: Springer, 2018. e-ISBN: 978-981-10-7739-5.
[4] STOLL Cliff. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*. Pocket Books, 2005, 399 s. ISBN 978-1416507789.
[5] COLLINS Michael. *Network Security Through Data Analysis: Building Situational Awareness*. O'Reilly Media, 2014, 348 s. ISBN 978-1449357900.
[6] MARAS Marie-Helen. *Computer Forensics: Cybercriminals, Laws, and Evidence*. 2nd ed. Jones & Bartlett Learning, 2014, 408 s. ISBN 978-1449692223.
[7] DIOGENES Yuri a OZKAYA, Erdal. *Cybersecurity: Attack and Defence Strategies*. Packt Publishing Limited, 2018, 449 s. ISBN 978-1788475297.
[8] SPITZNER Lance. Honeytokens: The Other Honeypot. In: *Broadcom* [online]. 16.června 2003 [cit. 2019-11-07]. Dostupné z: https://www.symantec.com/connect/articles/honeytokens-other-honeypot
[9] BERCOVITCH Maya, RENFORD Meir, HASSON Lior, SHABTAI Asaf, ROKACH Lior a ELOVICI Yuval. *HoneyGen: An automated honeytokens generator*. Beijing, China: IEEE, 2011. e-ISBN 978-1-4577-0085-9.
[10] Honeypot (computing). In: *SearchSecurity* [online]. Říjen 2018 [cit. 2019-11-07]. Dostupné z: https://searchsecurity.techtarget.com/definition/honey-pot
[11] CRIST Justin. Web Based Attacks. In: *Application and Database Security* [online]. SANS Institute, 2007, s. 4-5 [cit. 2019-11-10]. Dostupné z: https://www.sans.org/reading-room/whitepapers/application/paper/2053
[12] WATSON David. Low interaction honeypots revisited. *The honeynet project* [online]. [cit. 2019-11-26]. Dostupné z: https://www.honeynet.org/2015/08/06/low-interaction-honeypots-revisited/
[13] SANDERS Chris a SMITH Jason. *Applied Network Security Monitoring: Collection, Detection, and Analysis*. Syngress, 2013, 496 s. ISBN 97 8-0124172081.
[14] NIELS Provos. *A Virtual Honeypot Framework* [online]. [cit. 2019-12-26]. Dostupné z: https://www.usenix.org/legacy/publications/library/proceedings/sec04/tech/full_papers/provos/provos_html/honeyd.html
[15] LIVSHITZ Igor. What's the Difference Between a High Interaction Honeypot and a Low Interaction Honeypot? In: *Guardicore* [online]. [cit. 2019-12-26]. Dostupné z: https://www.guardicore.com/2019/1/high-interaction-honeypot-versus-low-interaction-honeypot/

[16] TAN Emil. Dionaea: A Malware Capturing Honeypot. In: *Division Zero* [online]. [cit. 2019-12-28]. Dostupné z: https://www.div0.sg/single-post/dionaea-malware-honeypot
[17] *GitHub: Kippo - SSH Honeypot* [online]. [cit. 2019-12-28]. Dostupné z: https://github.com/desaster/kippo
[18] *GitHub: Dockpot* [online]. [cit. 2020-01-03]. Dostupné z: https://github.com/eg-cert/dockpot
[19] HonSSH: Log all SSH communications between a client and server. In: *HonSSH* [online]. [cit. 2020-01-03]. Dostupné z: https://vulners.com/kitploit/KITPLOIT:2562697000631036022/
[20] ARTAIL Hassan, SAFA Haidar, SRAJ Malek, KUWATLY Iyad a AL-MASRI Zaid. A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks. *Computers & Security* [online]. Elsevier, 2006, 274-288 [cit. 2020-01-10]. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0167404806000587
[21] ZAKARIA Wira Zanoramy a KIAH Miss Laiha Mat. *A review of dynamic and intelligent honeypots* [online]. 2013 [cit. 2020-01-14]. DOI: 10.2306/scienceasia1513-1874.2013.39S.001. Dostupné z: http://scienceasia.org/2013.39S.n1/scias39S_1.pdf
[22] PROVOS Niels. A Virtual Honeypot Framework. In: *USENIX* [online]. 2004 [cit. 2020-01-15]. Dostupné z: https://www.usenix.org/legacy/publications/library/proceedings/sec04/tech/full_papers/provos/provos_html/honeyd.html
[23] PROVOS Niels a Thorsten HOLZ. *Virtual honeypots: from botnet tracking to intrusion detection*. Upper Saddle River, NJ: Addison-Wesley, 2008. ISBN 978-032-1336-323.
[24] DIKE Jeff. *User Mode Linux*. Upper Saddle River, NJ: Prentice Hall, 2006. ISBN 01-318-6505-6.
[25] KONIARIS Ioannis. Kippo-Graph. *BruteForce Lab's Blog: security, programming, devops, visualization, the cloud* [online]. [cit. 2020-03-07]. Dostupné z: https://bruteforce.gr/kippo-graph/
[26] What are the best SSH clients for Windows? In: *Slant* [online]. [cit. 2020-03-07]. Dostupné z: https://www.slant.co/topics/149/~best-ssh-clients-for-windows
[27] Best SSH Clients For Linux: Free And Paid SSH Tools. In: *Pickaweb* [online]. [cit. 2020-03-07]. Dostupné z: https://www.pickaweb.co.uk/kb/best-ssh-client-for-linux/
[28] HODGMAN, Roy. The Attacker's Dictionary. In: *Rapid7 Blog* [online]. 1.března 2016 [cit. 2020-03-07]. Dostupné z: https://blog.rapid7.com/2016/03/01/the-attackers-dictionary/
[29] *Honeypot as a Service* [online]. [cit. 2020-02-02]. Dostupné z: https://haas.nic.cz/
[30] SSH Honeypot – zajímavost. In: *Turris forum* [online]. 7.listopadu 2017 [cit. 2020-03-09]. Dostupné z: https://forum.turris.cz/t/ssh-honeypot-zajimavost/5584